

TUGAS AKHIR - KS 091336

**EVALUASI KUALITAS KEBUTUHAN
FUNGSIONAL DENGAN *FUNCTIONAL SIZE
MEASUREMENT* BERDASARKAN STANDAR
COSMIC ISO 19761 DAN METODE
REFACTORING (STUDI KASUS: WIKIBUDAYA)**

**PRIMA ARIFANDI
NRP 5210 100 049**

**Dosen Pembimbing
Feby Artwodini Muqtadiroh, S.Kom, M.T.
Amna Shifia Nisafani, S.Kom, M.Sc**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014**

FINAL PROJECT - KS 091336

**QUALITY EVALUATION OF FUNCTIONAL
REQUIREMENT USING FUNCTIONAL SIZE
MEASUREMENT BASED ON COSMIC ISO 19761
STANDARD AND REFACTORING METHOD
(CASE STUDY : WIKIBUDAYA)**

**PRIMA ARIFANDI
NRP 5210 100 049**

Supervisor

Feby Artwodini Muqtadiroh, S.Kom, M.T.

Amna Shifia Nisafani, S.Kom, M.Sc

**DEPARTMENT OF INFORMATION SYSTEM
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2014**

**EVALUASI KUALITAS KEBUTUHAN FUNGSIONAL
DENGAN FUNCTIONAL SIZE MEASUREMENT
BERDASARKAN STANDAR COSMIC ISO 19761 DAN
METODE REFACTORING
(STUDI KASUS: WIKIBUDAYA)**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer**

Pada

**Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya**

Oleh :

**Prima Arifandi
NRP. 5210 100 049**

Surabaya, Juli 2014

JURUSAN SISTEM INFORMASI

**Dr. Eng. Febridiyan Samopa, S.Kom., M.Kom
NIP. 19730219 199802 1 001**

**EVALUASI KUALITAS KEBUTUHAN FUNGSIONAL
DENGAN FUNCTIONAL SIZE MEASUREMENT
BERDASARKAN STANDAR COSMIC ISO 19761 DAN
METODE REFACTORING
(STUDI KASUS: WIKIBUDAYA)**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada**

**Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya**

Oleh :

**Prima Arifandi
NRP. 5210 100 049**

**Disetujui Tim Penguji : Tanggal Ujian : 11 Juli 2014
Periode Wisuda : September**

Febby Artwodini Muqtadiroh, S.Kom, M.T..... (Pembimbing 1)

Amna Shifia Nisafani, S.Kom, M.Sc..... (Pembimbing 2)

Sholih, S.T., M.Kom, M.SA..... (Penguji 1)

Dr. Apol Pribadi, S.T., M.T..... (Penguji 2)

**EVALUASI KUALITAS KEBUTUHAN FUNGSIONAL
DENGAN FUNCTIONAL SIZE MEASUREMENT
BERDASARKAN STANDAR COSMIC ISO 19761 DAN
METODE REFACTORING (STUDI KASUS:
WIKIBUDAYA)**

Nama Mahasiswa : Prima Arifandi
NRP : 5210 100 049
Jurusan : Sistem Informasi FTIF-ITS
Dosen : Feby Artwodini M., S.Kom, M.T.
Pembimbing : Amna Shifia Nisafani, S.Kom, M.Sc

Abstrak

Perangkat lunak yang berkualitas adalah kondisi dimana sistem, komponen atau proses dapat memenuhi ekspektasi pengguna. Kualitas perangkat lunak sangatlah penting, karena dapat mempengaruhi kepuasan pengguna, niat untuk menggunakan dan keuntungan. Oleh karena itu, hal tersebut sangat diperlukan untuk mempertahankan tingkat kualitas sebuah perangkat lunak dengan cara meningkatkan kualitasnya. Penelitian ini bertujuan untuk meningkatkan kualitas kebutuhan fungsional WikiBudaya dalam dua cara, yaitu menemukan kecacatan dengan menggunakan Functional Size Measurement berdasarkan standar ISO 19761 dan memperbaiki kecacatan yang terdeteksi dengan menggunakan metode refactoring. Output dari penelitian ini adalah sebuah laporan SKPL (Spesifikasi Kebutuhan Perangkat Lunak) dari WikiBudaya yang telah diperbaharui sesuai dengan hasil identifikasi kecacatan dan metode refactoring.

Kata Kunci : Software quality, COSMIC ISO 19761, refactoring, functional requirement, WikiBudaya

**QUALITY IMPROVEMENT OF FUNCTIONAL
REQUIREMENT USING FUNCTIONAL SIZE
MEASUREMENT BASED ON COSMIC ISO 19761
STANDARDS AND REFACTORING METHOD (CASE
STUDY: WIKIBUDAYA)**

Student Name : Prima Arifandi
NRP : 5210 100 049
Majority : Sistem Informasi FTIF-ITS
Supervisor : Feby Artwodini M., S.Kom, M.T.
Amna Shifia Nisafani, S.Kom, M.Sc

Abstract

A high quality software is a condition in which a system, component or process can meet user expectations. Software quality is very important since it may affect user satisfaction, intention to use and net benefit. Therefore, it is indispensable to maintain software quality level by means of improving its quality. Here, this study aims to improve the quality of functional requirement of WikiBudaya in twofold: finding defects using functional size measurement based on the COSMIC ISO 19761 standard, and refining the detected defects using refactoring method. The output of this research is a SRS (Software Requirement Spesification) report of WikiBudaya that has been refined accordance with the defect identification and refactoring method.

Keywords : Software quality, COSMIC ISO 19761, refactoring, functional requirement, WikiBudaya

KATA PENGANTAR

Segala puji dan syukur penulis tuturkan kehadirat Allah SWT yang telah memberikan kekuatan dan kesempatan untuk penulis sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul:

EVALUASI KUALITAS KEBUTUHAN FUNGSIONAL DENGAN FUNCTIONAL SIZE MEASUREMENT BERDASARKAN STANDAR COSMIC ISO 19761 DAN METODE REFACTORING (STUDI KASUS: WIKIBUDAYA)

Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang telah meluangkan waktu, tenaga dan pikirannya bagi penulis untuk menyelesaikan tugas akhir ini. Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak di bawah ini, yaitu:

1. Ibu Feby Artwodini Muqtadiroh, S.Kom, M.T., selaku dosen pembimbing I yang telah meluangkan waktu dan pikiran beliau untuk membimbing dan mengarahkan penulis dalam pengerjaan tugas akhir ini.
2. Ibu Amna Shifia Nisafani, S.Kom, M.Sc selaku dosen pembimbing II yang telah meluangkan waktu dan pikiran beliau untuk membimbing dan mengarahkan penulis dalam pengerjaan tugas akhir ini.
3. Bapak Toni Dwi Susanto, S.T, M.T., Ph.D selaku dosen penguji I yang telah meluangkan waktunya untuk menguji dan memberikan masukan serta kritik yang membangun dalam pengerjaan tugas akhir ini.
4. Bapak Dr. Apol Pribadi Subriadi, S.T, M.T., selaku dosen penguji II yang telah meluangkan waktunya untuk menguji dan memberikan masukan serta kritik yang membangun dalam pengerjaan tugas akhir ini.

5. Bapak Ir. Achmad Holil Noor Ali, M.Kom selaku dosen wali yang selalu memberikan masukan kepada penulis pada saat perwalian dalam mengambil mata kuliah.
6. Anggota laboratorium PPSI yang tidak bisa disebutkan satu persatu dan tak henti – hentinya memberikan dukungan kepada penulis.
7. FOXIS, angkatan 2010 Jurusan Sistem Informasi ITS yang selalu memberikan dukungannya, dan membantu penulis dalam menjalani perkuliahan.
8. Kepada seluruh angkatan di Jurusan Sistem Informasi yang telah banyak mengajarkan banyak hal kepada penulis.
9. Anggota WB_Crew, Geng Guritasisasi dan BAB yang tidak bisa disebutkan satu persatu dan tak henti – hentinya memberikan support kepada penulis,
10. Serta seluruh pihak yang telah membantu penulis dalam mengerjakan tugas akhir ini yang tidak mungkin disebutkan satu per satu.

Penulis pun sadar bahwa tugas akhir ini masih belum sempurna dengan segala kekurangan di dalamnya. Oleh karena itu penulis memohon maaf atas segala kekurangan yang ada pada tugas akhir ini. Penulis sangatlah senang apabila ada pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, Juli 2014

Penulis

DAFTAR ISI

Abstrak	ix
Abstract	xi
KATA PENGANTAR	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Tugas Akhir.....	4
1.5 Manfaat Kegiatan Tugas Akhir	5
1.6 Sistematika Pengerjaan Tugas Akhir.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 <i>Software Quality</i>	7
2.2 <i>Software Requirement</i>	9
2.3 <i>Functional User Requirement</i>	10
2.4 <i>Functional Size Measurement</i>	12
2.5 Cosmic ISO 19761	12
2.6 <i>Use Case</i>	16
2.7 <i>Use Case Scenario</i>	16
2.8 <i>Refactoring</i>	17
2.9 WikiBudaya.....	18
BAB 3 METODE PENELITIAN.....	21
3.1 Measurement Strategy	22
3.2 Mapping Phase	23
3.3 Measurement Phase.....	24
3.4 Proses Refactoring.....	24
3.5 Kesimpulan dan Saran.....	25
3.6 Penyusunan Laporan Tugas Akhir	26
BAB 4 HASIL DAN PEMBAHASAN	33
4.1 Measurement Strategy	33

4.1.1	<i>Requirement Definition Artifacts</i>	33
4.1.2	<i>Data analysis</i>	37
4.1.3	<i>Dynamic Behaviour Models</i>	40
4.1.4	Penentuan Software Context Model	44
4.2	Mapping Phase.....	45
4.3	Measurement Phase	51
4.4	Proses Refactoring.....	62
4.5	Proses Re-Measurement Phase	91
BAB 5	101
PENUTUP	101
5.1	Kesimpulan.....	101
5.2	Saran	102
BAB 6	DAFTAR PUSTAKA	103
	BIODATA PENULIS	105
	LAMPIRAN A	A-1
	LAMPIRAN B	B-1
	LAMPIRAN C	C-1
	LAMPIRAN D	D-1

DAFTAR TABEL

Tabel 2.1 Tipe Kecacatan [13]	15
Tabel 3.1 Rangkuman Metodologi Penelitian	27
Tabel 4.1 Keterangan Proses Bisnis 1	35
Tabel 4.2 Keterangan Proses Bisnis 2	35
Tabel 4.3 Daftar Use Case	36
Tabel 4.4 Penyusunan Kebutuhan Fungsional Baru	41
Tabel 4.5 Software Context Model	44
Tabel 4.6 Penjabaran Proses Fungsional	45
Tabel 4.7 UC-01.01 Daftar Akun	47
Tabel 4.8 UC-01.02 Masuk Akun	49
Tabel 4.9 UC-01.03 Keluar Akun	50
Tabel 4.10 UC-01.04 Lihat Detail Akun Pribadi	50
Tabel 4.11 UC-01.05 Ubah Detail Akun Pribadi	51
Tabel 4.12 Analisa Cacat UC-01.01	52
Tabel 4.13 Analisa Cacat UC-01.02	54
Tabel 4.14 Analisa Cacat UC-01.03	55
Tabel 4.15 Analisa Cacat UC-01.04	55
Tabel 4.16 Tabel Hasil Analisa Kecacatan Use Case	55
Tabel 4.17 Tabel Presentase Kesesuaian Use Case	60
Tabel 4.18 Tabel identifikasi peluang & solusi Refactoring	62
Tabel 4.19 UC-01.01 sebelum refactoring	66
Tabel 4.20 UC-01.01 setelah refactoring	68
Tabel 4.21 UC-01.02 sebelum refactoring	70
Tabel 4.22 UC-01.02 setelah refactoring	71
Tabel 4.23 UC-01.05 sebelum refactoring	72
Tabel 4.24 UC-01.05 setelah refactoring	73
Tabel 4.25 UC-01.07 sebelum refactoring	74
Tabel 4.26 UC-01.07 setelah refactoring	74
Tabel 4.27 UC-01.09 sebelum refactoring	75
Tabel 4.28 UC-01.09 setelah refactoring	76
Tabel 4.29 UC-01.10 sebelum refactoring	76
Tabel 4.30 UC-01.10 setelah refactoring	77

Tabel 4.31 UC-01.11 sebelum refactoring	77
Tabel 4.32 UC-01.11 setelah refactoring.....	77
Tabel 4.33 UC-02.02 sebelum refactoring	78
Tabel 4.34 UC-02.02 setelah refactoring.....	79
Tabel 4.35 UC-02.03 sebelum refactoring	80
Tabel 4.36 UC-02.03 setelah refactoring.....	80
Tabel 4.37 UC-02.04 sebelum refactoring	81
Tabel 4.38 UC-02.04 setelah refactoring.....	82
Tabel 4.39 UC-02.05 sebelum refactoring	83
Tabel 4.40 UC-02.05 setelah refactoring.....	83
Tabel 4.41 UC-02.06 sebelum refactoring	84
Tabel 4.42 UC-02.06 setelah refactoring.....	85
Tabel 4.43 UC-02.09 sebelum refactoring	86
Tabel 4.44 UC-02.09 setelah refactoring (Move Activity).....	86
Tabel 4.45 UC-02.09 setelah refactoring (Extract Alternatvie Flows).....	87
Tabel 4.46 UC-02.12 sebelum refactoring	88
Tabel 4.47 UC-02.12 setelah refactoring.....	88
Tabel 4.48 UC-03.01 sebelum refactoring	88
Tabel 4.49 UC-03.01 setelah refactoring.....	89
Tabel 4.50 UC-03.03 sebelum refactoring	89
Tabel 4.51 C-03.03 setelah refactoring	90
Tabel 4.52 Re-Measurement Phase UC Daftar Akun Kontributor	91
Tabel 4.53 Re-Measurement Phase UC Masuk Akun	93
Tabel 4.54 Re-Measurement Phase UC Ubah Detail Akun Pribadi	94
Tabel 4.55 Use Case hasil dari proses refactoring.....	95
Tabel 4.56 Penjabaran KF-01 baru.....	96
Tabel 4.57 Penjabaran KF-02 baru.....	96
Tabel 4.58 Penjabaran KF-03 baru.....	97

DAFTAR GAMBAR

Gambar 2.1 Tabel Kebutuhan Fungsional [11]	11
Gambar 2.2 Proses pengukuran COSMIC ISO-19761 [10]	13
Gambar 2.3 Alur pergerakan data [10].....	14
Gambar 2.4 Contoh Use Case dan Aktor [14].....	16
Gambar 2.5 Use Case Scenario [14].....	17
Gambar 3.1 Flow Chart Pengerjaan Tugas Akhir	21
Gambar 3.2 Proses penentuan FUR	22
Gambar 4.1 Proses Bisnis WikiBudaya.....	34
Gambar 4.2 CDM WikiBudaya.....	38
Gambar 4.3 PDM WikiBudaya	39
Gambar 4.4 Layer WikiBudaya.....	42
Gambar 4.5 Dekomposisi KF-01.....	43
Gambar 4.6 Dekomposisi KF-02.....	43
Gambar 4.7 Dekomposisi KF-03.....	44
Gambar 4.8 Gambar Kecacatan Berdasarkan Literatur [13]	97
Gambar 4.9 Klasifikasi kecacatan berdasarkan peluang refactoring	98

Halaman ini sengaja dikosongkan.

BAB 1

PENDAHULUAN

Dalam bab ini membahas tentang latar belakang pengerjaan tugas akhir, rumusan permasalahan yang dihadapi dalam pengerjaan tugas akhir, batasan permasalahan pengerjaan tugas akhir, tujuan pengerjaan tugas akhir, dan manfaat dari pengerjaan tugas akhir.

1.1 Latar Belakang

Pada perancangan sebuah perangkat lunak, kualitas perangkat lunak memegang peranan penting dalam penilaian perangkat lunak tersebut. Menurut [1] kualitas perangkat lunak adalah kajian turun temurun dalam sejarah ilmu rekayasa perangkat lunak (*software engineering*). Kajian ini dimulai dari apa yang akan diukur (apakah proses atau produk), apakah memang perangkat lunak bisa diukur, sudut pandang pengukur dan bagaimana menentukan parameter pengukuran kualitas perangkat lunak. Kualitas perangkat lunak akan mempengaruhi kepuasan pelanggan, niat untuk menggunakan dan keuntungan bagi perusahaan. Maka dari itu, perlu dilakukan suatu peningkatan kualitas sebuah perangkat lunak. Agar dapat tercipta perangkat lunak yang memenuhi ketiga aspek yang telah disebutkan sebelumnya.

Melalui pengujian kualitas perangkat lunak, dapat ditemui kecacatan yang terdapat pada suatu perangkat lunak. Kecacatan pada perangkat lunak tersebut merupakan salah satu penyebab buruknya kualitas suatu perangkat lunak, adapun perangkat lunak yang memiliki kualitas yang buruk adalah perangkat lunak yang tidak sesuai dengan ekspektasi pengguna. Kecacatan pada perangkat lunak bisa saja ditemukan pada bagian kebutuhan perangkat lunaknya, yaitu kebutuhan fungsional dan kebutuhan non-fungsionalnya [2].

Studi kasus yang diangkat kali ini adalah situs WikiBudaya. Situs ini merupakan sebuah situs yang berisi tentang kebudayaan yang

ada di Indonesia dan dikelola oleh Dinas Kebudayaan Dan Pariwisata Provinsi Jawa Timur. Umumnya, situs ini memiliki fungsi sama seperti situs – situs ensiklopedia yang ada pada masa kini. Situs WikiBudaya memberikan informasi tentang kebudayaan yang ada di seluruh Indonesia. Metode yang digunakan dalam pengembangan situs WikiBudaya ini adalah *Incremental Model*. Pada proses *Incremental Model*, tim pengembang situs WikiBudaya hanya berfokus pada analisis dan desain situs WikiBudaya dan diselesaikan dalam kurun waktu tiga bulan.

Peneliti menemukan bahwa masih ada pengembangan sebuah aplikasi sistem informasi yang menghasilkan aplikasi yang tidak berkualitas. Ada banyak factor yang menyebabkan aplikasi yang dibangun tidak berkualitas diantaranya adalah kebutuhan pengguna yang sering berubah. Namun pada paper [3] menemukan adanya isu kurang baik yang sangat mempengaruhi kualitas pengembangan suatu perangkat lunak, yaitu pada proses identifikasi kebutuhan (Requirements). Buruknya kualitas pada tahap identifikasi kebutuhan akan menyebabkan kegagalan dalam mengetahui aspek fungsionalitas dan kebutuhan pengguna. Contohnya adalah identifikasi kebutuhan dalam skala besar yang mengakibatkan kurangnya kejelasan terhadap aspek fungsionalitas yang terdapat pada aplikasi sistem informasi tersebut yang berdampak pada munculnya lazy requirements, adanya aktivitas yang sama dalam satu proses (duplikat), dan sebagainya. Permasalahan ini lah yang sering ditemui dalam aktivitas identifikasi kebutuhan dalam proses pengembangan sistem informasi. [4]

Pada penelitian kali ini, peningkatan kebutuhannya akan difokuskan pada kebutuhan fungsional saja tidak sampai pada kebutuhan non-fungsional. Karena, metode yang dipakai pada penelitian ini hanya berfokus pada kebutuhan fungsional suatu sistem saja.

Peningkatan kualitas kebutuhan fungsional situs WikiBudaya akan dilakukan melalui pengukuran *functional size* berdasarkan standar COSMIC ISO 19761 dan metode *refactoring*. COSMIC ISO 19761 merupakan sebuah metode tentang pengukuran kualitas atau ukuran dari kebutuhan fungsional sebuah perangkat lunak yang didukung dan diakui oleh standar internasional ISO [5]. Melalui pengukuran *functional size* berdasarkan standar COSMIC ISO 19761 ini akan ditemukan suatu kecacatan pada kebutuhan fungsional situs WikiBudaya. Kecacatan inilah yang menyebabkan suatu perangkat lunak memiliki kualitas yang buruk. Maka dari itu akan dilakukan pula perbaikan pada kecacatan tersebut. Sedangkan metode yang akan digunakan untuk memperbaiki kecacatan tersebut adalah dengan menggunakan metode *refactoring*. *Refactoring* menurut [3] adalah sebuah operasi restrukturisasi program yang bertujuan untuk mendukung desain dan evolusi dari sebuah perangkat lunak yang berorientasi objek, dengan cara meningkatkan perancangan perangkat lunak tersebut dengan menggunakan transformasi sistematis

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan pada sub bab sebelumnya, permasalahan yang akan diangkat dalam tugas akhir ini adalah:

1. Apakah terdapat kecacatan pada kebutuhan fungsional situs WikiBudaya.
2. Apa sajakah kecacatan yang terdapat pada kebutuhan fungsional situs WikiBudaya.
3. Perbaikan apa saja yang perlu dilakukan dalam rangka meningkatkan kualitas kebutuhan fungsional situs WikiBudaya.

1.3 Batasan Masalah

1. Dokumen pendukung yang digunakan adalah dokumen tugas akhir “*Analisa Dan Perancangan WikiBudaya dalam Rangka Melestarikan Budaya Bangsa Dan Kearifan Lokal Nusantara*” oleh Nanda Fitrianto Nugraha, Institut Teknologi Sepuluh Nopember, tahun 2013 dan “*Pembangunan Perangkat Lunak Wikibudaya Untuk Melestarikan Budaya Dan Kearifan Lokal Indonesia*” oleh Mochammad Misbach Arifianto, Institut Teknologi Sepuluh Nopember, tahun 2013.
2. Nilai *cosmic functional point* (cfp) yang akan muncul setelah dilakukan pengukuran functional size berdasarkan standar ISO COSMIC 19761 tidak akan digunakan, karena tidak akan dilakukan pengukuran estimasi biaya.
3. Pengklasifikasian kecacatan yang digunakan dalam pengerjaan tugas akhir ini sesuai dengan dokumen “*Classification of Defect Types in Requirements Specifications: Literature Review, Proposal and Assessment*” oleh Isabel Lopes Margarido dan “*Improving the Quality of Requirements with Refactoring*” oleh Ricardo Ramos
4. Kebutuhan fungsional yang telah ditingkatkan kualitasnya dan diperbaiki kecacatannya akan disajikan dalam bentuk laporan SKPL (Spesifikasi Kebutuhan Perangkat Lunak).

1.4 Tujuan Tugas Akhir

Tujuan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Untuk mengetahui kecacatan yang terdapat pada kebutuhan fungsional situs WikiBudaya.
2. Untuk memperbaiki kebutuhan fungsional dari situs WikiBudaya.
3. Untuk meningkatkan kualitas kebutuhan fungsional dari situs WikiBudaya.

1.5 Manfaat Kegiatan Tugas Akhir

Manfaat dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Mempermudah tim pengembang untuk mengembangkan WikiBudaya kedepannya, karena informasi tentang kebutuhan sistem yang lebih jelas dan lebih detail.
2. Memberikan pemahaman lebih detail pada tahap kebutuhan pengembangan WikiBudaya.
3. Memberikan saran kepada tim pengembang untuk pengembangan WikiBudaya kedepannya.

1.6 Sistematika Pengerjaan Tugas Akhir

Sistematika penulisan atau pengerjaan tugas akhir ini adalah sebagai berikut:

1 BAB I

PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan permasalahan, batasan masalah/ruang lingkup, tujuan, manfaat dan sistematika penulisan.

2 BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan istilah-istilah yang digunakan pada penulisan buku tugas akhir ini serta dasar teori yang digunakan pada tugas akhir ini.

3 BAB III

METODE PENELITIAN

Bab ini membahas alur dan tata pengerjaan tugas akhir dari awal sampai selesainya tugas akhir.

4 BAB IV

HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil dari pengerjaan tugas akhir. Yaitu keluaran dari setiap tahap dalam metode yang telah dijelaskan.

5 BAB V

PENUTUP

Bab ini menjelaskan kesimpulan dari pengerjaan tugas akhir ini, serta saran yang membangun tentang aplikasi yang berhubungan dengan tugas akhir ini.

6 BAB VI

Daftar Pustaka

Bab ini berisi sumber data-data yang dijadikan acuan pembelajaran dalam pengerjaan tugas akhir.

BAB 2

TINJAUAN PUSTAKA

Tinjauan pustaka merupakan penjelasan mengenai teori-teori terkait bersumber dari buku, jurnal, artikel, ataupun tugas akhir terdahulu yang berfungsi sebagai dasar dalam melakukan pengerjaan tugas akhir agar dapat memahami konsep atau teori penyelesaian permasalahan yang ada.

2.1 Konsep Kualitas

Menurut [6] konsep kualitas sering dianggap sebagai ukuran relatif kebaikan sebuah produk barang atau jasa yang terdiri dari kualitas desain dan kualitas kesesuaian. Kualitas desain merupakan fungsi spesifikasi produk, sedangkan kualitas kesesuaian adalah suatu ukuran tentang seberapa jauh suatu produk mampu memenuhi persyaratan atau spesifikasi kualitas yang telah ditetapkan.

Pada [6] juga didefinisikan penjelasan lain akan kualitas, yaitu sebagai kondisi dinamis yang berhubungan dengan produk, jasa, manusia, proses dan lingkungan yang memenuhi atau melebihi harapan. Mengamati kedua definisi tersebut terlihat bahwa walaupun terdapat perbedaan, namun secara implisit juga terdapat kesamaan. Kesamaan tersebut terletak pada konsepsi kualitas sebagai kondisi yang dapat memenuhi apa yang seharusnya. Sedangkan berikut merupakan tujuh poin tentang konsep kualitas[6] , yaitu:

1. Kesesuaian dengan persyaratan dan tuntutan;
2. Kecocokan untuk pemakaian;
3. Perbaikan atau penyempurnaan yang berkelanjutan;
4. Bebas dari kerusakan atau cacat;
5. Pemenuhan kebutuhan pelanggan semenjak awal dan setiap saat
6. Melakukan sesuatu secara benar semenjak awal;
7. Sesuatu yang membahagiakan pelanggan.

2.2 *Software Quality*

Menurut [7] *Software quality* adalah kesesuaian secara eksplisit yang menyatakan bahwa persyaratan fungsional dan kinerja pengembangan perangkat lunak telah sesuai dengan harapan dan telah dikembangkan secara profesional. Pada intinya, *software* dapat dikatakan berkualitas apabila sesuai dengan kebutuhan yang telah ditentukan. Adapula jaminan tentang kualitas perangkat lunak tersebut biasa disebut dengan SQA atau *Software Quality Assurance*. SQA merupakan suatu pola yang terencana dan sistematis tentang tindakan yang diperlukan untuk menyediakan kepercayaan bahwa suatu perangkat lunak tersebut telah sesuai dengan yang diharapkan. SQA sendiri juga merupakan kegiatan yang dirancang untuk mengevaluasi proses dimana perangkat lunak tersebut dikembangkan [7]. Berikut merupakan beberapa penyebab kesalahan pada sebuah perangkat lunak [7] :

1. Definisi akan persyaratan rusak.
Kesalahan pada tipe ini lebih disebabkan kepada pengguna. Pengguna tidak sepenuhnya mengetahui bahkan tidak mengetahui bagaimana seharusnya perangkat lunak yang diinginkan berjalan sesuai kebutuhan.
2. Kegagalan komunikasi antara klien dengan tim pengembang.
Kegagalan dalam mengidentifikasi kebutuhan perangkat lunak biasanya disebabkan oleh komunikasi yang buruk antara klien dengan pengembang. Komunikasi yang buruk salah satunya adalah kurang tanggapnya seorang pengembang terhadap perubahan kebutuhan perangkat lunak yang disampaikan oleh klien, yang berakibat pada kualitas perangkat lunak yang akan dibangun nantinya.
3. Penyimpangan yang disengaja.
Dalam beberapa situasi tertentu, biasanya pengembang tidak mengikuti persyaratan yang telah disepakati sebelumnya yang mengakibatkan kesalahan dalam pengembangan perangkat lunak. Pengembang biasanya menggunakan modul pada proyek sebelumnya untuk kemudian diterapkan pada proyek

pengembangan perangkat lunak yang sedang dijalankan. Tanpa ada komunikasi lebih lanjut, hal ini akan mempengaruhi kualitas dari perangkat lunak yang akan dibangun.

4. Kesalahan dalam desain logis.
Kesalahan pada perangkat lunak bisa terjadi pada saat *design* sebuah perangkat lunak dibuat oleh mereka yang kurang memiliki kompeten di bidangnya. Kesalahan biasanya terjadi dalam proses mendefinisikan alur system yang akan dibuat.
5. Penulisan kode pemrograman yang tidak sesuai.
Kesalahan pada saat penulisan baris kode pemrograman adalah ketika seorang programmer kurang memahami desain dari perangkat lunak dan pada akhirnya terjadi ketidaksesuaian antara kode pemrograman dengan desain sehingga hal ini akan mempengaruhi kualitas dari output yang akan dihasilkan dari perangkat lunak.
6. Ketidakesesuaian pada dokumentasi dan kode pemrograman.
Pengembang biasanya memiliki standar dan prosedur tersendiri dalam mengembangkan sebuah perangkat lunak. Namun dalam sebuah proyek pengembangan perangkat lunak, hal ini justru akan menjadi hal yang kurang baik untuk jangka panjang pada saat melakukan perbaikan karena ketidaksesuaian antara dokumentasi dan kode pemrograman antara pengembang dengan tim proyek.
7. Kesalahan pada dokumentasi.
Kesalahan pada dokumentasi akan mempersulit proses pengembangan dan pemeliharaan perangkat lunak yang dilakukan oleh tim pemeliharaan perangkat lunak. Kesalahan yang terjadi berupa kurangnya integrasi antara dokumen kebutuhan perangkat lunak dengan desain dokumen yang berakibat pada pengembangan perangkat lunak selanjutnya.

2.3 Software Requirement

Kebutuhan perangkat lunak adalah kondisi yang harus dimiliki untuk memenuhi apa yang disyaratkan oleh perangkat lunak atau

diinginkan oleh pemakai. Analisis kebutuhan adalah proses untuk menetapkan fungsi perangkat lunak, menyatakan antarmuka perangkat lunak dengan elemen-elemen sistem lain, dan menentukan kendala yang harus dihadapi oleh perangkat lunak [7]. Adapula beberapa kesalahan yang sering terjadi dalam proses menganalisa kebutuhan [8], berikut:

1. Pelanggan yang tidak mengerti apa yang sebenarnya mereka inginkan.
2. Perubahan kebutuhan pada saat proyek telah berlangsung.
3. Pelanggan yang memiliki *timeline* yang tidak masuk akal.
4. Adanya pembeda yang menyebabkan hubungan antar pelanggan, teknisi dan *project manager* terhambat.
5. Tim pengembang yang tidak mengetahui kebijakan politik dari perusahaan tempat pelanggan bekerja.

Dalam perancangan perangkat lunak, kebutuhan itu sendiri terdiri dari kebutuhan fungsional, kebutuhan non-fungsional, dan batasan/*constraint*. Kebutuhan fungsional mendeskripsikan kemampuan dan layanan dari sebuah sistem. Kebutuhan non-fungsional mendeskripsikan tingkatan dari kualitas, misalnya seberapa aman, dapat digunakan atau tidak, dan lainnya. *Constraint* adalah kondisi yang tidak dinegosiasikan yang mempengaruhi sebuah proyek, misalnya tingkat *skill* dari tim pengembang, budget yang tersedia, waktu pengerjaan proyek, atau infrastruktur komputer pada lingkungan pengembangan [9]. Adapula rekayasa kebutuhan atau juga disebut dengan *requirements engineering*, adalah cabang dari rekayasa perangkat lunak yang fokus pada tujuan, fungsi atau kegunaan, dan batasan pada sistem perangkat lunak. Selain itu juga pada hubungan antara faktor-faktor tadi untuk menentukan spesifikasi dari sebuah perangkat lunak dan beradaptasi terhadap perkembangan dari waktu ke waktu [9].

2.4 Functional User Requirement

Functional User Requirement atau FUR adalah sebuah *subset* dari kebutuhan *user* yang menggambarkan apa saja yang harus dilakukan oleh sebuah perangkat lunak [10]. Dalam rekayasa perangkat lunak, *Functional User Requirement* bertugas mendefinisikan fungsi dari suatu sistem atau komponen pembangun perangkat lunak, fungsi ini digambarkan sebagai satu set input, perilaku dan output. Maka dari itu, kebutuhan fungsional menentukan hal tertentu dari suatu sistem. Kebutuhan fungsional sendiri berhubungan dengan beberapa hal sebagai berikut:

1. *Transfer* data, yang merupakan pergerakan suatu data dari pengguna menuju ke sistem atau sebaliknya.
2. Penyimpanan data, yang merupakan pergerakan suatu data dari sistem menuju ke *data storage*.
3. Pengambilan data, yang merupakan pergerakan suatu data dari *data storage* menuju sistem.

Berikut Gambar 2.1 yang menunjukkan contoh dari kebutuhan fungsional:

ID KF	Deskripsi KF	ID UC	Deskripsi UC
KF01	Useradmin melakukan login	UC01	Useradmin dapat melakukan login
KF02	Kasir melakukan login	UC02	Kasir dapat melakukan login
KF03	User melakukan login	UC03	User melakukan login
KF04	Mengelola data pengguna sistem	UC04.a	Menambah data pengguna sistem
		UC 04.b	Mengubah data pengguna sistem
		UC 04.c	Menghapus data pengguna sistem
KF05	Mengelola halaman informasi	UC 05.a	Menambah informasi
		UC 05. b	Mengubah informasi
		UC 05. c	Menghapus informasi

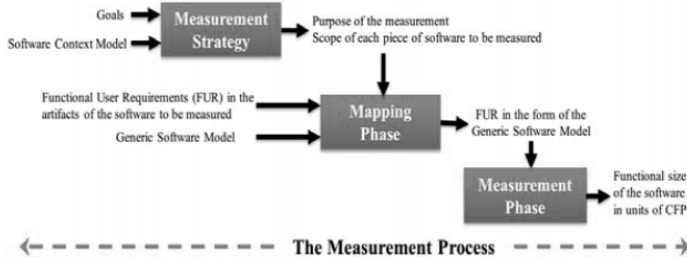
Gambar 2.1 Tabel Kebutuhan Fungsional [11]

2.5 *Functional Size Measurement*

Functional Size Measurement atau FSM adalah metode untuk mengukur fungsionalitas suatu perangkat lunak. Sedangkan menurut [12], fungsinya adalah untuk memonitor progres dan performa, menentukan nilai produktifitas secara keseluruhan, perencanaan yang lebih baik dan kepastian untuk mendapatkan pengukuran yang berkualitas tinggi. Ukuran fungsional sendiri dapat digunakan untuk berbagai keperluan, namun hal itu digunakan terutama pada tahap perencanaan sebagai *input* ke dalam perhitungan estimasi biaya suatu *project resource*. FSM juga memungkinkan digunakan untuk membandingkan aplikasi dan proyek berdasarkan ukuran mereka. Tingkat produktivitas suatu aplikasi dari atribut profil yang sama dapat dibandingkan untuk pembandingan tujuan perbaikan. Tingkat suatu produktivitas dari proyek yang sudah lama juga dapat digunakan untuk memprediksi usaha, walaupun ukuran fungsional proyek tersebut telah ditentukan.

2.6 *Cosmic ISO 19761*

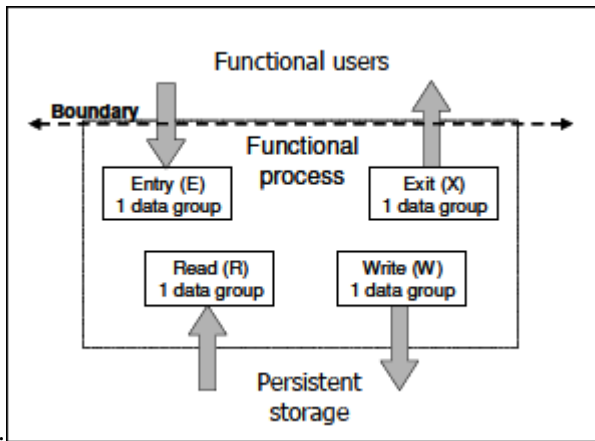
COSMIC merupakan kependekan dari *The Common Software Measurement International Consortium*. COSMIC bertujuan untuk menguji, mengembangkan dan mencari penerimaan baru dari suatu perangkat lunak baru dengan menggunakan metode pengukuran estimasi dan kinerja [10]. COSMIC sendiri didirikan pada tahun 1998. Sedangkan menurut [5] adalah metodologi FSM generasi kedua yang dikembangkan oleh COSMIC yang masuk dalam standard ISO (10761) EIC-2009 dirancang untuk mengukur *functional size* dari sistem informasi manajemen, *software realtime* and sistem *multi-layer*. Di dalam COSMIC ISO 19761, terdapat tiga tahap proses pengukuran, yaitu *measurement stratgy*, *mapping phase* dan *measurement phase* [10]. Penjelasan mengenai tahap – tahap tersebut dapat dilihat pada Gambar 2.2 :



Gambar 2.2 Proses pengukuran COSMIC ISO-19761 [10]

Dari Gambar 2.2 pada tahap awal atau *measurement strategy*, *measurer* atau seseorang yang melakukan pengukuran dengan metode ini akan merumuskan empat hal penting, yaitu tujuan melakukan pengukuran, ruang lingkup pengukuran, identifikasi kebutuhan pengguna dan tingkat keakuratan. Sedangkan pada tahap *mapping phase* pengguna akan merumuskan tentang identifikasi kebutuhan fungsional, identifikasi grup data dan identifikasi atribut dari data tersebut. Pada tahap akhir pengukuran akan dilakukan penghitungan ukuran *cfp* atau *COSMIC functional point* dan pengidentifikasian kecacatan.

Pengukuran dengan standar COSMIC ISO 19761 akan difokuskan pada kebutuhan fungsional suatu perangkat lunak. Setelah didapatkan kumpulan kebutuhan fungsional dari suatu perangkat lunak, maka akan diukur nilai fungsionalnya dan akan diidentifikasi kecacatannya. Satuan nilai fungsional didapatkan dengan menghitung berapa jumlah pergerakan data yang terdapat pada satu proses fungsional. Nilai fungsional tersebut disajikan dalam satuan cfp (*COSMIC Functional Point*) [10]. Penjelasan dan detail dari pergerakan data pada suatu proses fungsional dapat dilihat pada Gambar 2.3:



Gambar 2.3 Alur pergerakan data [10]

1. *Entry (E)* : pergerakan grup data dari luar (*boundary*) menuju *function process*.
2. *Exit (X)* : pergerakan grup data dari *functional process* menuju luar (*boundary*).
3. *Read (R)* : pergerakan grup data dari tempat penyimpanan menuju *function process*.
4. *Write (W)* : pergerakan grup data dari *functional process* menuju tempat penyimpanan

Setelah pergerakan data ditentukan, maka selanjutnya adalah proses penghitungan jumlah unit *cfp*-nya, proses ini dilakukan dengan menggunakan rumus (1) berikut:

$$\begin{aligned}
 &Size\ CFP(functional\ process_{i_j}) \\
 &= \sum size(Entries_{i_j}) \\
 &+ \sum size(exists_{i_j}) + \sum size(Reads_{i_j}) + \sum size(Writes_{i_j}) \dots\dots(1)
 \end{aligned}$$

Dengan menggunakan rumus tersebut, dapat ditentukan berapa total *cosmic functional point* dalam satu proses fungsional, yaitu dengan menjumlahkan seluruh pergerakan data yang terdapat dalam satu proses fungsional.

Sedangkan untuk identifikasi kecacatan yang terdapat pada suatu FUR perangkat lunak pada tahap *measurement phase* akan diketahui dengan mengikuti indikator berikut [13]:

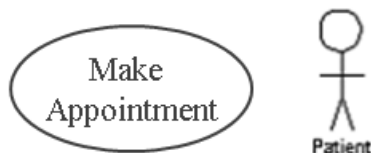
Tabel 2.1 Tipe Kecacatan [13]

Tipe Kecacatan	Penjelasan
<i>Missing or Incomplete</i>	Terdapat informasi yang tidak disebutkan pada dokumen, informasi yang relevan hilang, sehingga menyebabkan kebutuhan ini tidak lengkap.
<i>Incorrect Information</i>	Terdapat informasi yang salah atau informasi yang diberikan bertentangan dengan kebutuhan yang dimaksud.
<i>Inconsistent</i>	Terdapat informasi yang tidak konsisten dengan dokumen atau memiliki konflik dengan kebutuhan lainnya yang sudah benar.
<i>Ambiguous or Unclear</i>	Terdapat informasi atau sebuah kosa kata yang mempunyai lebih dari satu interpretasi.
<i>Misplaced</i>	Terdapat informasi yang berada tidak pada tempatnya, baik dalam bagian spesifikasi kebutuhan, fungsionalitas atau sistem yang mengacu.
<i>Infeasible or Non-verifiable</i>	Terdapat informasi yang tidak dapat diimplementasikan karena terbatasnya teknologi.
<i>Redundant or Duplicate</i>	Terdapat pengulangan informasi atau kebutuhan yang sudah terdapat pada

	kebutuhan yang lainnya.
<i>Typo or Formatting</i>	Ortografi, semantik atau kesalahan tata bahasa maupun pengejaan yang salah pada sebuah kata. Aturan format yang salah juga termasuk dalam kategori ini.
<i>Not Relevant or Extraneous</i>	Terdapat informasi atau bagian dari spesifikasi kebutuhan yang diluar dari <i>scope</i> proyek yang berhubungan atau terdapat informasi yang tidak diperlukan.

2.7 Use Case

Use Case adalah sebuah cara formal untuk mewakili bagaimana sebuah sistem bisnis berinteraksi dengan lingkungannya [14]. *Use case* sendiri digunakan untuk membentuk tingkah laku sesuatu dalam sebuah sistem serta direalisasikan oleh seorang aktor (*user*). Pada umumnya *use case* digambarkan dalam bentuk elips dan memiliki sebuah nama. Sedangkan aktor adalah suatu bentuk yang mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem [14]. Aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick man*.



Gambar 2.4 Contoh Use Case dan Aktor [14]

2.8 Use Case Scenario

Use case scenario atau *skenario use case* adalah sebuah model yang menggambarkan bagaimana suatu sistem bekerja secara keseluruhan dan pada skenario tertentu [14]. Sehingga jika sebuah objek merupakan sesuatu yang menyediakan layanan, maka harus dapat dibuktikan bahwa semua layanan tersebut ada dalam suatu use case. Adapun contoh dari *use case scenario* dapat dilihat pada Gambar 2.5

Nama Use-case:	<i>Make Appointment</i>
Aktor:	<i>Patient dan scheduler</i>
Deskripsi:	Membuat temu janji untuk berobat
Normal Course:	1. <i>Patient</i> menginisialisasi proses ini dengan membuat temu janji untuk berobat dengan mendatangi tempat praktek. 2. Hal ini kemudian dicatat oleh <i>scheduler</i> .
Alternate Course:	1a. <i>Patient</i> dapat membuat janji lewat telepon. 2a. <i>Scheduler</i> dapat menuliskan temu janji berdasarkan informasi yang diberikan oleh <i>patient</i> .
Pre-Condition:	<i>Patient</i> sakit
Post-Condition:	<i>Cancel appointment, check patient's record, request medication</i>
Assumption:	-

Gambar 2.5 Use Case Scenario [14]

2.9 Refactoring

Refactoring merupakan sebuah operasi restrukturisasi program yang bertujuan untuk mendukung desain dan evolusi dari sebuah perangkat lunak yang berorientasi objek, dengan cara meningkatkan perancangan perangkat lunak tersebut dengan menggunakan transformasi sistematis [3]. Pada paper [3] juga dijelaskan mengenai peluang refactoring yang digunakan untuk setiap kesempatan yang telah ditetapkan, berikut penjabarannya:

1. Large Requirement

Suatu kondisi dimana sebuah use case mencoba untuk mengakomodasi beberapa fungsi / tujuan sekaligus atau sebuah use case yang memiliki alternative flows dan langkah - langkah yang berlebihan.

2. *Complex Conditional Structure*

Suatu kondisi dimana sebuah use case kebutuhan perangkat lunak memiliki struktur yang kompleks atau kebutuhan perangkat lunak tersebut membutuhkan beberapa kebutuhan perangkat lunak lain agar menjadi satu kesatuan kebutuhan perangkat lunak yang baik.

3. *Lazy Requirement*

- a. Suatu kondisi dimana fungsi / peran sebuah kebutuhan perangkat lunak memiliki impact yang tidak jelas terhadap sistem.
- b. Suatu kondisi dimana sebuah kebutuhan perangkat lunak tidak mengakomodasi seluruh aktivitas yang dimaksud / incomplete requirements.

4. *Naming Problems*

Suatu kondisi dimana sebuah penamaan dalam kebutuhan perangkat lunak tidak mengacu pada konsep yang telah ditentukan atau suatu kondisi dimana penamaan yang sama digunakan untuk konsep yang berbeda (Ambigu).

5. *Duplicated Activities*

Suatu kondisi dimana sebuah kebutuhan perangkat lunak yang sama memiliki duplikat pada tempat yang berbeda pada dokumen kebutuhan perangkat lunak. Salah satu contohnya adalah dimana sebuah *main flow* atau *alternative flow* diulang pada sebuah kebutuhan perangkat lunak.

2.10 WikiBudaya

WikiBudaya adalah sebuah situs yang membahas tentang kebudayaan yang ada di Indonesia, situs ini sendiri dikelola oleh Dinas Kebudayaan Dan Pariwisata Provinsi Jawa Timur [15]. Umumnya, situs ini memiliki fungsi sama seperti situs-situs ensiklopedia yang ada pada masa kini. Situs WikiBudaya memberikan informasi tentang kebudayaan yang ada di seluruh Indonesia, tetapi WikiBudaya memiliki para reviewer yang merupakan seorang ahli budaya dan dipilih sendiri oleh Dinas

Kebudayaan Dan Pariwisata Provinsi Jawa Timur dari komunitas - komunitas budaya yang ada di Indonesia.

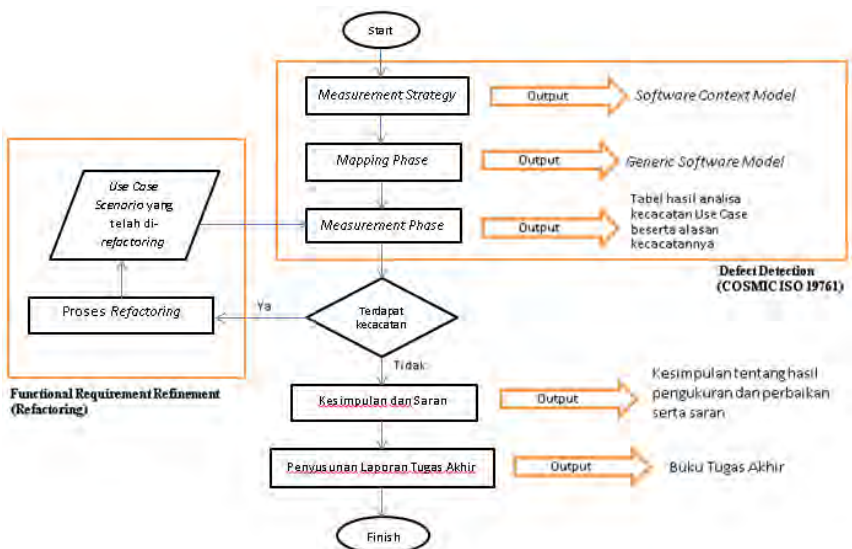
Untuk mengakses WikiBudaya, nantinya *user* dapat melakukan dengan dua langkah. Langkah pertama *user* dapat melakukan pencarian artikel budaya, dengan mengetikkan kata kunci dari artikel budaya yang ingin diketahui. Adapun langkah pencarian artikel budaya dilakukan dengan masuk (*login*) akun terlebih dahulu, untuk memiliki akun WikiBudaya *user* diharuskan untuk melakukan pendaftaran akun terlebih dahulu. Selanjutnya *user* dapat melakukan pencarian dengan mengetikkan kata kunci dari artikel budaya yang diinginkan, dan sistem akan menampilkan hasil pencarian artikel budaya. Pada artikel tersebut terdapat kolom untuk memberikan komentar terkait artikel tersebut, tetapi sebelum memberikan komentar *user* diwajibkan untuk melakukan login akun wikibudaya terlebih dahulu. Jika *user* belum memiliki akun maka akan diarahkan untuk mendaftar akun terlebih dahulu. Berikut merupakan aktor yang terdapat pada situs WikiBudaya [15] :

1. Administrator : Administrator dari wikibudaya ada dua yaitu admin yang bertindak sebagai admin sistem dan admin reviewer.
 - a. Admin Sistem : Bertugas untuk menjaga keamanan dari sistem wikibudaya, menghapus komentar yang dianggap tidak layak, menghapus akun yang tidak aktif, dan menerima pengguna baru dari wikibudaya. Bagian dari disbudpar yang di tunjuk sebagai admin sistem adalah bagian seksi data dan informasi.
 - b. Admin Reviewer : Bertugas sebagai koordinator dari reviewer, sehingga bagi pengguna (baik umum dan komunitas) yang ingin menjadi reviewer nantinya akan divalidasi terlebih dahulu oleh admin reviewer, mengenai kelayakan untuk menjadi seorang reviewer. Bagian seksi budaya yang akan bertanggung jawab sebagai admin reviewer.

2. Reviewer : Merupakan sekumpulan orang yang memiliki tugas untuk melakukan review dan validasi artikel budaya. Setiap artikel yang dirasa layak oleh reviewer maka, dapat ditampilkan pada wikibudaya. Untuk menjadi reviewer maka, diharuskan untuk mendaftar akun sebagai reviewer yang nantinya akan divalidasi oleh admin reviewer terlebih dahulu.
3. Kontributor : Orang-orang yang memiliki rasa untuk melestarikan budaya bangsa, dengan ikut berkontribusi mengirimkan artikel-artikel budaya yang diketahuinya. Untuk menjadi seorang kontributor diharuskan melakukan masuk (login) akun sebagai kontributor terlebih dahulu.
4. User-umum : Merupakan masyarakat umum secara luas, yang dapat mengakses wikibudaya tanpa perlu melakukan pendaftaran atau masuk (login) akun terlebih dahulu.

BAB 3 METODE PENELITIAN

Bab Metode Penelitian ini menjelaskan tentang alur pengerjaan tugas akhir. Metode pengerjaan tugas akhir digunakan sebagai acuan agar pengerjaan tugas akhir terarah dan jelas. Berikut Gambar 3.1 yang merupakan *flowchart* dari pengerjaan tugas akhir ini.



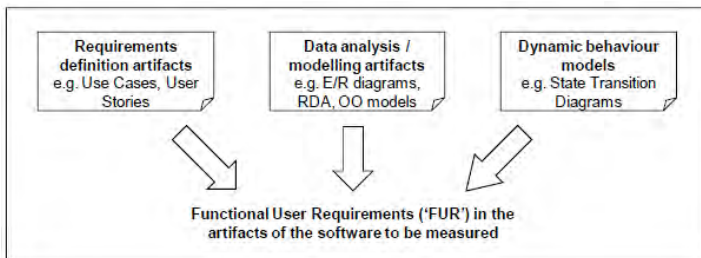
Gambar 3.1 Flow Chart Pengerjaan Tugas Akhir

Pengerjaan tugas akhir ini berfokus pada pencarian kecacatan dan perbaikan kecacatan dari kebutuhan fungsional WikiBudaya. Tahapan pengerjaan tugas akhir ini dilakukan dengan menggunakan standar COSMIC ISO 19761 dan metode *refactoring*. Tahap pencarian kecacatan sendiri akan dilakukan dengan standar COSMIC ISO 19761 yang terdiri dari tiga tahapan, yaitu *measurement strategy*, *mapping phase* dan *measurement phase*. Sedangkan setelah kecacatan tersebut

ditemukan, maka akan diperbaiki dengan menggunakan metode *refactoring*.

3.1 Measurement Strategy

Tahap *Measurement Strategy* merupakan tahap awal dalam pengerjaan tugas akhir ini. Pada tahap ini terdapat beberapa proses yang akan dilakukan, yaitu *Requirement Definition Artifacts*, *Data analysis* dan *Dynamic Behaviour Models*. Proses ini bertujuan untuk menentukan *Functional User Requirement* yang terdapat pada WikiBudaya. Setelah dilakukan beberapa proses tersebut, maka dapat ditentukan output dari tahap ini, yaitu sebuah *Software Context Model*, yang merupakan sebuah tabel yang berisi tujuan dari pengukuran, ruang lingkup pengukuran dan pengguna fungsional dari WikiBudaya. Berikut merupakan Gambar 3.2 yang merupakan gambaran dari beberapa proses yang terdapat pada tahap ini :



Gambar 3.2 Proses penentuan FUR

Pada tahap *Requirement Definition Artifacts* akan dilakukan pengidentifikasian kebutuhan perangkat lunak WikiBudaya sesuai dengan cerita dari pengguna dan proses bisnis yang telah didefinisikan pada dokumen [15].

Tahap *Data analysis* merupakan tahap penentuan struktur model dari WikiBudaya. Struktur model dari WikiBudaya didapat dari *Entity Relationship Diagram* yang telah didefinisikan pada

dokumen [15]. Dari dokumen [15] tersebut akan didapatkan dua model *Entity Relationship Diagram*, yaitu CDM (Conceptual Data Model) dan PDM (Physician Data Model).

Pada tahap *Dynamic Behaviour Models* akan ditentukan pengguna fungsional dari WikiBudaya. Sesuai dengan proses bisnis yang telah didefinisikan pada dokumen [15], maka dapat diidentifikasi pengguna fungsional dari WikiBudaya.

Setelah cerita pengguna, proses bisnis, ER *diagram* dan pengguna fungsional didapatkan, maka *Fuctional User Requirement* dapat diidentifikasi. Setelah didapatkan *Functional User Requirement* dari WikiBudaya, maka tahap selanjutnya adalah pembuatan pemetaan dekomposisi dari WikiBudaya. Hal ini dilakukan agar dapat diketahui dengan jelas *layer* apa saja yang terdapat pada WikiBudaya. Setelah dilakukan pemetaan dekomposisi tersebut, maka selanjutnya akan dilakukan pula pemetaan dekomposisi terhadap kebutuhan fungsional WikiBudaya.

Serangkaian proses diatas dilakukan untuk dapat melakukan penentuan *Software Context Model*, yang merupakan output dari tahap ini yang berupa suatu tabel yang menjelaskan tentang tujuan pengukuran, ruang lingkup pengukuran dan *functional user* dari WikiBudaya.

3.2 Mapping Phase

Mapping phase merupakan tahap kedua dari COSMIC ISO 19761. Pada tahap ini akan diidentifikasi *sub-process* dari *Functional User Requirement* yang telah teridentifikasi pada tahap sebelumnya. Sedangkan kumpulan *sub-process* tersebut telah terdefiniskan di dalam dokumen [15] . Maka langkah selanjutnya adalah membuat tabel *Generic Software Model*, yang merupakan output dari tahap ini. *Generic Software Model* merupakan suatu tabel yang berisi kumpulan proses fungsional dan *sub-process* dari proses fungsional WikiBudaya.

3.3 Measurement Phase

Pada tahap ini akan dilakukan pengukuran *cosmic functional point* pada setiap proses fungsional dan *sub-process* dari proses fungsional situs WikiBudaya. Jika pada proses pengukuran ini terdapat sebuah *sub-process* yang memiliki nilai pergerakan data "0" maka sudah dipastikan *use case* tersebut memiliki kecacatan, sehingga harus diperbaiki. Serta pada tahap ini pula akan dilakukan proses analisa kecacatan pada setiap proses fungsional yang memiliki kecacatan. Penentuan tipe kecacatan terhadap proses fungsional yang memiliki kecacatan tersebut berdasarkan pada klasifikasi kecacatan yang berdasarkan pada dokumen [13]. Sedangkan *output* dari tahap ini adalah sebuah tabel hasil analisa kecacatan Use Case beserta alasan kecacatannya.

3.4 Proses Refactoring

Proses *refactoring* hanya dilakukan pada setiap kebutuhan fungsional yang memiliki kecacatan saja. Tetapi sebelum dilakukan proses *refactoring*, akan dilakukan terlebih dahulu proses analisa peluang dan solusi *refactoring* terhadap *use case* yang memiliki kecacatan. Setelah itu baru akan dilakukan perbaikan terhadap *use case* tersebut. Ada lima cara yang dapat dilakukan untuk mengatasi kecacatan tersebut atau solusi *refactoring* [3], berikut:

1. *Extract Requirement*

Proses ini dilakukan apabila ada informasi kebutuhan perangkat lunak dalam skala besar yang dapat dibagi menjadi dua atau lebih sebagai kebutuhan yang baru.

2. *Rename Requirements*

Pemberian nama pada kebutuhan perangkat lunak disesuaikan dengan konteks dari kebutuhan perangkat lunak tersebut. Pemberian nama yang baik akan mempermudah komunikasi dan pemahaman terhadap abstraksi sistem.

3. ***Move Activity***

Proses ini dilakukan untuk meningkatkan modularitas dan menyeimbangkan aktivitas antara kebutuhan perangkat lunak. Proses ini bisa juga terjadi pada saat proses ekstraksi kebutuhan (*Extract Requirement*) dengan memindahkan aktivitas ke kebutuhan yang diinginkan.

4. ***Inline Requirement***

Proses ini dilakukan untuk mengurangi kompleksitas dari kebutuhan perangkat lunak dengan cara menggabungkan beberapa kebutuhan perangkat lunak yang ada. Jika sebuah kebutuhan perangkat lunak tidak cukup penting untuk digunakan, maka pengembang dapat melakukan *inline*, *merging* terhadap kebutuhan perangkat lunak yang lain.

5. ***Extract Alternative Flows***

Proses ini dilakukan pada saat *information flow* pada kebutuhan perangkat lunak melakukan beberapa skenario dalam waktu yang bersamaan sehingga terjadi penumpukan informasi yang berakibat pada pemahaman *requirement responsibilities* dan *information flow* yang minim dan sulit.

Setelah proses *refactoring* selesai, maka akan dihasilkan sebuah kebutuhan fungsional yang tidak memiliki kecacatan, lalu kebutuhan fungsional tersebut akan diukur ulang dengan menggunakan standar COSMIC ISO 19761 pada tahap *Measuring Phase*.

3.5 **Kesimpulan dan Saran**

Pada tahap ini akan dilakukan pembuatan kesimpulan dari hasil pengukuran dan perbaikan kebutuhan fungsional dari situs WikiBudaya, sehingga akan diketahui kebutuhan fungsional yang telah diperbaiki dari hasil *refactoring*. Serta, pada tahap ini juga akan diberikan saran berupa pengembangan atau perbaikan untuk pengembangan situs WikiBudaya selanjutnya.

3.6 Penyusunan Laporan Tugas Akhir

Tahap ini dilakukan selama masa penelitian. Setiap langkah-langkah pengerjaan tugas akhir ini dari awal hingga akhir didokumentasikan dan ditulis dalam sebuah laporan yang sesuai dengan format buku tugas akhir.

Untuk memperjelas metodologi penelitian yang digunakan untuk pengerjaan tugas akhir, perhatikan tabel berikut :

Tabel 3.1 Rangkuman Metodologi Penelitian

Fase	Aktivitas	Tujuan	Input	Output	Standart
<i>Defect Detection</i>	<i>Measurement Strategy</i>	Perumusan tiga hal penting (tujuan melakukan pengukuran, ruang lingkup pengukuran dan identifikasi pengguna fungsional)	<i>Software Artifact</i>	<i>Software Context Model</i> <i>(tabel yang berisi tujuan pengukuran, ruang lingkup pengukuran dan pengguna fungsional)</i>	COSMIC ISO 19761

	<i>Mapping Phase</i>	Menentukan proses fungsional dan <i>sub-process</i> dari proses fungsional	<i>Software Context Model</i>	<i>Generic Software Model</i> (tabel yang berisi kumpulan proses fungsional dan <i>sub-process</i>)
	<i>Measurement Phase</i>	Pengukuran <i>cosmic functional point</i> dan identifikasi kecacatan proses fungsional	<i>Generic Software Model</i>	Tabel hasil analisa kecacatan Use Case beserta alasan kecacatannya

<i>Functional Requirement Refinement</i>	<i>Extract Requirement</i>	Memecah <i>use case</i> yang sudah ada menjadi dua atau lebih <i>use case</i> yang baru	Adanya <i>use case</i> yang memiliki banyak aktivitas dan memiliki <i>output</i> yang berlebihan	Dua atau lebih <i>use case</i> yang aktivitasnya telah disesuaikan	<i>Refactoring</i>
	<i>Rename Requirement</i>	Pemberian nama yang baik untuk membuat pemahaman dan komunikasi menjadi lebih mudah	Adanya penamaan suatu <i>use case</i> / aktor yang tidak sesuai	<i>Use case</i> atau aktor yang telah diubah namanya sesuai aktivitasnya	

	<i>Move Activity</i>	Memindahkan aktivitas dalam suatu <i>use case</i> ke <i>use case</i> yang lain	Adanya aktivitas yang dapat dipindah ke <i>use case</i> yang lain	Aktivitas yang sudah dipindahkan ke <i>use case</i> yang sesuai
	<i>Inline Requirement</i>	Menyisipkan <i>use case</i> yang sudah ada pada <i>use case</i> yang lain	Adanya <i>use case</i> yang dapat disisipkan dengan <i>use case</i> yang lain	<i>Use case</i> baru yang telah disisipkan pada sebuah <i>use case</i> lainnya
	<i>Extract Alternative Flows</i>	Memecah <i>main flows</i> dalam suatu <i>use case</i> karena dapat dipecah menjadi <i>alternative</i>	Adanya <i>alternative flows use case</i> yang menghasilkan banyak <i>output</i>	<i>Alternative flows</i> dari suatu <i>use case</i> yang telah disesuaikan

		<i>flows atau exception</i>			
--	--	---------------------------------	--	--	--

Halaman ini sengaja dikosongkan.

BAB 4

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas hasil pengerjaan tugas akhir, yaitu keluaran dari setiap tahap dalam metode yang telah dijelaskan pada bab sebelumnya, berikut:

4.1 Measurement Strategy

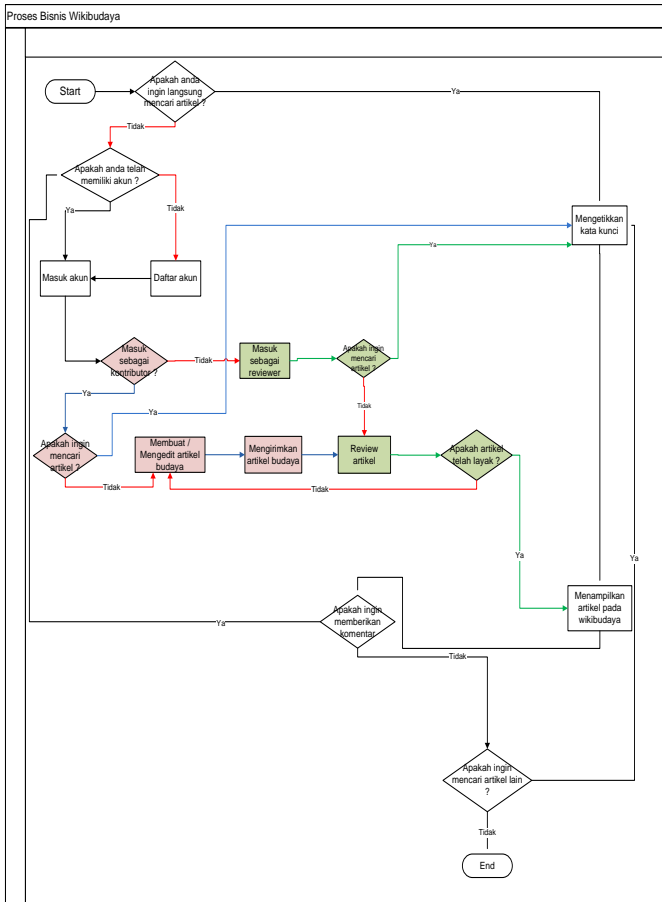
Pada tahap awal pengidentifikasian kecacatan dengan menggunakan metode *measurement strategy*, dalam tahap ini dilakukan proses *Requirement Definition Artifacts*, *Data analysis* dan *Dynamic Behaviour Models* yang akan menghasilkan beberapa output yang digunakan sebagai acuan dalam menentukan *output* dari tahap awal ini, yaitu sebuah tabel *Software Context Model*.

4.2 Requirement Definition Artifacts

Di bawah ini merupakan hasil dari tahapan *requirement definiton artifacts*, yaitu sebuah kumpulan dari cerita pengguna dan tabel proses bisnis WikiBudaya yang didapatkan dari dokumen [15]. Berikut merupakan kumpulan cerita pengguna yang didapat dari dokumen [15] :


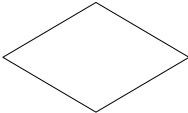

1. Menginginkan adanya pemberian informasi akan kebudayaan nusantara dengan menggunakan teknologi informasi.
2. Menginginkan adanya sistem yang membahas kebudayaan nusantara tetapi dari sumber yang terpercaya.
3. Sistem tentang kebudayaan nusantara yang nantinya akan dibangun diharapkan menggunakan perangkat lunak berbasis web, yang dapat diakses kapan pun dan dimana pun.

Sedangkan berikut merupakan tabel proses bisnis yang tergambarkan pada dokumen [15] :









Gambar 4.1 Proses Bisnis WikiBudaya

Tabel 4.1 Keterangan Proses Bisnis 1

Simbol	Nama	Keterangan
	Terminator	Menunjukkan awal atau akhir dari aliran proses
	Decision	Decision digunakan untuk melambangkan pengambilan keputusan bagaimana alur dalam flow chart berjalan selanjutnya berdasarkan kriteria atau pertanyaan tertentu
	Process	Untuk menunjukkan sebuah proses atau operasi

Tabel 4.2 Keterangan Proses Bisnis 2

Warna	Keterangan
	Untuk alur “ Tidak”
	Untuk “ reviewer”
	Untuk “ kontributor”
	Untuk alur “reviewer”
	Untuk alur “ Kontributor”
	Untuk alur yang dapat diakses semua pengguna

Dari cerita pengguna dan tabel proses bisnis WikiBudaya yang telah terdefinisi pada dokumen [15] , maka dengan begitu dapat diidentifikasi kebutuhan apa saja yang dibutuhkan pengguna pada WikiBudaya nantinya, berikut merupakan kumpulan *use case* dari WikiBudaya yang telah ditetapkan dan telah terdefinisi pada dokumen [15] :

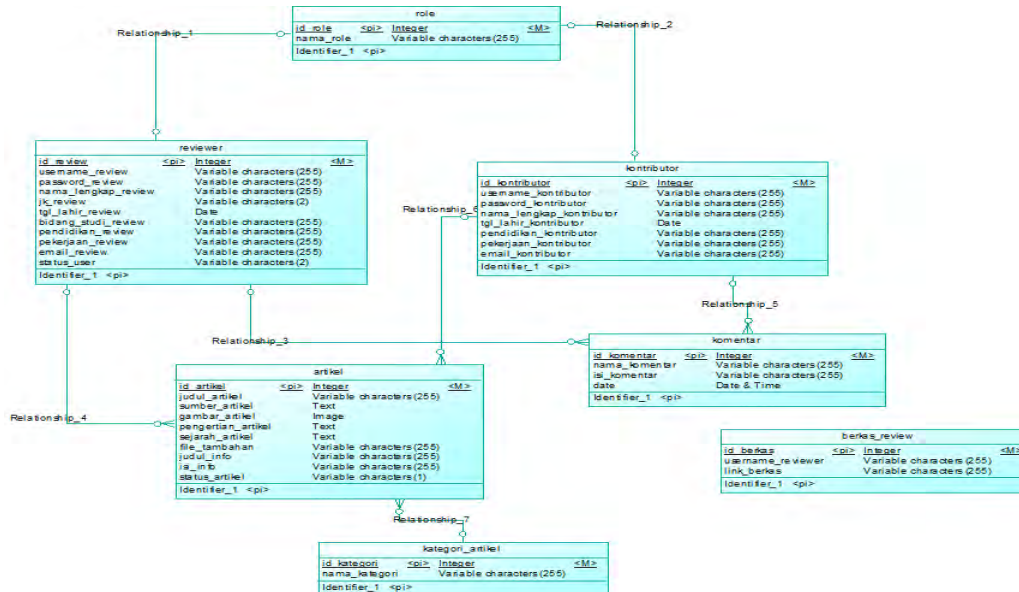
Tabel 4.3 Daftar Use Case

No.	Nama <i>Use Case</i>
1	Daftar Akun
2	Masuk Akun
3	Keluar Akun
4	Lihat detail akun pribadi
5	Ubah detail akun pribadi
6	Lihat Data User
7	Hapus Akun
8	Cari User
9	Validasi Kontributor
10	Validasi Reviewer
11	Ubah Kata Kunci
12	Lihat data master artikel budaya
13	Tambah artikel budaya
14	Ubah artikel budaya
15	Hapus artikel budaya
16	Cari artikel budaya
17	Unggah file (lagu,video)
18	Unduh file (lagu,video)
19	Review Artikel
20	Validasi artikel
21	Lihat artikel
22	Lihat Status
23	Lihat Histori Artikel
24	Beri Komentar
25	Lihat Komentar
26	Hapus Komentar
27	Notifikasi komentar

4.3 Data analysis

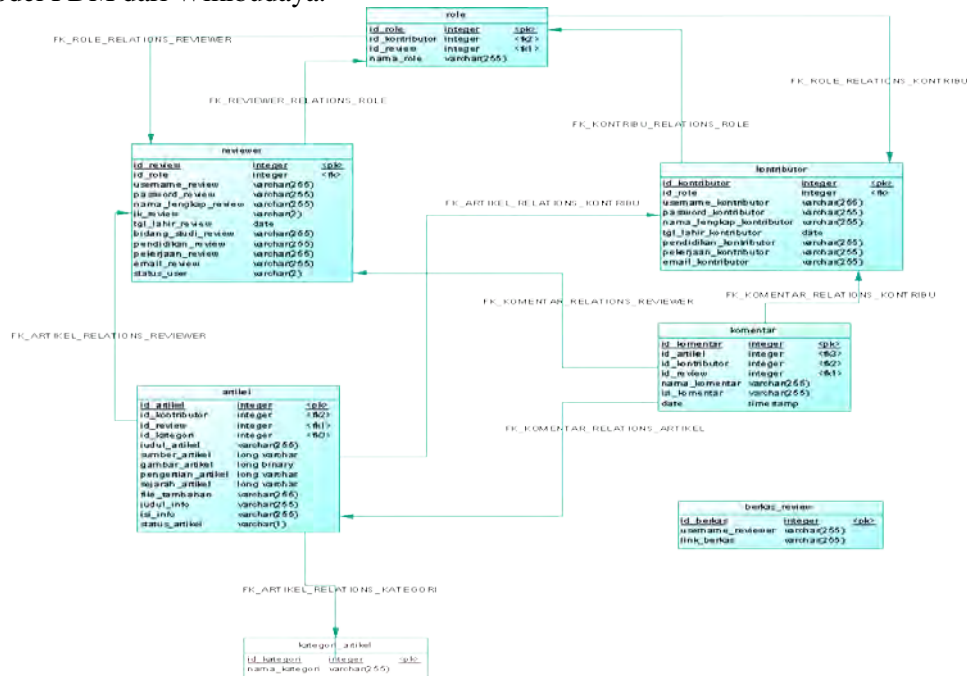
Di bawah ini merupakan hasil dari tahapan *data analysis*, yaitu sebuah ER *Diagram* atau *Entity Relationship Diagram* WikiBudaya yang didapatkan dari dokumen [15]. Berikut merupakan dua model ER *Diagram* yang digunakan, yaitu CDM (*Conceptual Data Model*) dan PDM (*Physician Data Model*) yang didapat dari dokumen [15] :

Berikut model CDM dari Wikibudaya:



Gambar 4.2 CDM WikiBudaya

Berikut model PDM dari Wikibudaya:



Gambar 4.3 PDM WikiBudaya

4.4 Dynamic Behaviour Models

Berdasarkan pada proses bisnis dan kumpulan *use case* yang telah teridentifikasi pada tahap sebelumnya, maka dengan begitu dapat diketahui pengguna fungsional dari WikiBudaya, berikut merupakan daftar pengguna fungsional WikiBudaya berdasarkan dokumen [15] :

1. **Admin Sistem:** bertugas menjaga keamanan dan mengelola sistem dari WikiBudaya.
2. **Admin Reviewer:** bertugas sebagai koordinator dari reviewer.
3. **Reviewer:** bertugas melakukan review dan validasi artikel budaya.
4. **Kontributor:** berkontribusi mengirimkan artikel - artikel budaya.
5. **User-Umum:** masyarakat umum yang dapat mengakses WikiBudaya tanpa perlu login.

Maka dengan ditemukannya kumpulan *use case* dan daftar pengguna fungsional WikiBudaya, sesuai dokumen [15], berikut merupakan daftar *Functional User Requirement* dari WikiBudaya:

- | | |
|-------|--|
| KF-01 | Sistem menyediakan fitur bagi pengguna untuk melakukan daftar (<i>register</i>) akun, masuk (<i>login</i>) akun, dan keluar (<i>logout</i>) akun |
| KF-02 | Sistem menyediakan fitur bagi admin (reviewer & sistem) untuk mengelola master, termasuk didalamnya adalah data master akun, artikel, dan komentar |
| KF-03 | Sistem menyediakan fitur bagi pengguna (aktor) selain admin untuk mengelola akun |
| KF-04 | Sistem dapat menampilkan history dari artikel yang pernah kirimkan oleh kontributor dan di review oleh reviewer. |

- KF-05 Sistem menyediakan fitur unggah dan unduh file budaya.
- KF-06 Sistem menyediakan fitur bagi admin reviewer untuk melakukan review dan validasi artikel
- KF-07 Sistem dapat menampilkan komentar-komentar dari setiap artikel budaya
- KF-08 Sistem menyediakan fitur untuk memberikan komentar

Tetapi setelah diidentifikasi ulang, ternyata kebutuhan fungsional yang didapat dari dokumen [15] masih terlalu luas dan rancu. Jadi kebutuhan fungsional diatas akan dipersempit lagi berdasarkan aturan *SMART Requirements* sesuai dengan dokumen [16]. Aturan ini digunakan untuk mendeteksi kerancuan yang ada pada pernyataan spesifikasi kebutuhan perangkat lunak. Sesuai dengan aturan tersebut, maka sebaiknya suatu kebutuhan fungsional suatu perangkat lunak haruslah spesifik, yaitu suatu pernyataan kebutuhan yang ditulis secara jelas, konsisten, sederhana dan tepat. Maka dari itu, berikut merupakan tabel yang menunjukkan pembentukan kebutuhan fungsional baru dari WikiBudaya :

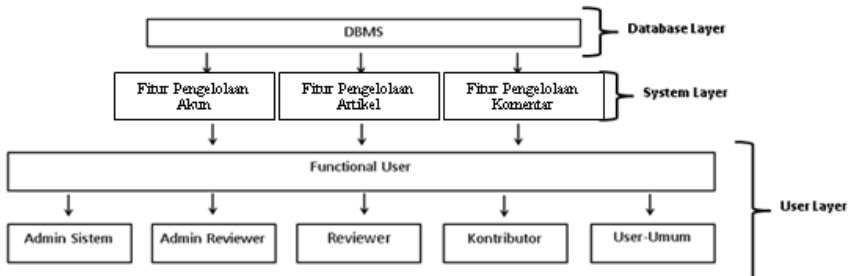
Tabel 4.4 Penyusunan Kebutuhan Fungsional Baru

Kebutuhan Fungsional Baru	Tersusun dari
KF-01 (Pengelolaan Akun)	KF-01
	KF-02
	KF-03
KF-02 (Pengelolaan Artikel)	KF-04
	KF-05
	KF-06
KF-03 (Pengelolaan Komentar)	KF-07
	KF-08

Berdasarkan tabel diatas, maka didapatkan hasil kebutuhan fungsional baru dari WikiBudaya yang lebih spesifik dan lebih jelas. Berikut merupakan kebutuhan fungsional baru tersebut yang telah sesuai aturan *SMART requirements* :

- KF-01 Sistem menyediakan fitur untuk mengelola akun.
- KF-02 Sistem menyediakan fitur untuk mengelola artikel.
- KF-03 Sistem menyediakan fitur untuk mengelola komentar.

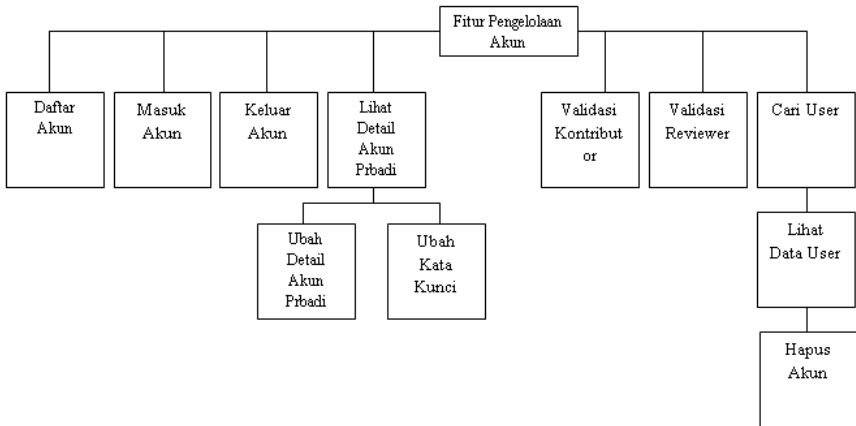
Dari hasil pembaharuan terhadap kebutuhan fungsional WikiBudaya, maka kebutuhan fungsional tersebut terlihat lebih spesifik dan jelas, sehingga memudahkan dalam memetakan *use case* apa saja yang terdapat pada tiap-tiap kebutuhan fungsional tersebut. Selanjutnya akan dilakukan proses pemetaan berdasar *layer* atau lapisan yang terdapat pada WikiBudaya, berikut hasil dari pemetaan tersebut :



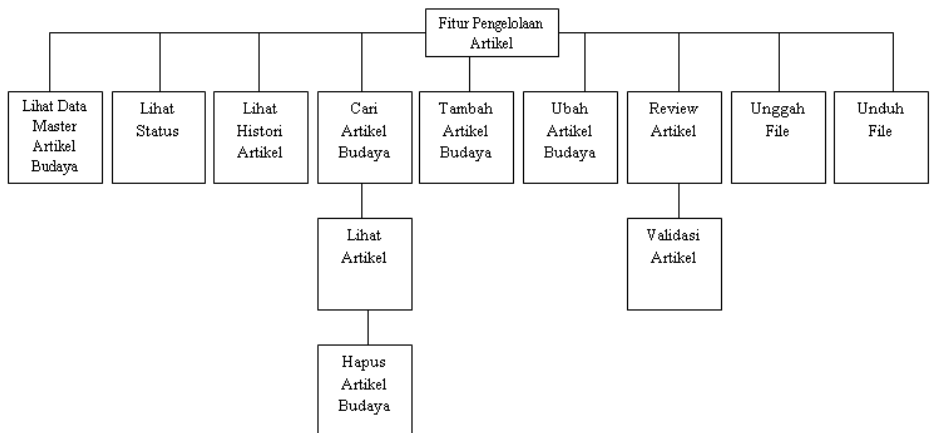
Gambar 4.4 Layer WikiBudaya

Dari Gambar 4.10 tersebut, dapat terlihat terdapat 3 *layer* pada WikiBudaya, yaitu *database layer*, *system layer*, dan *user layer*. Di dalam *database layer* diketahui merupakan lapisan dimana *database* WikiBudaya bekerja untuk menyalin dan membaca data, sedangkan dalam *system layer* merupakan tempat dimana ketiga kebutuhan fungsional WikiBudaya bekerja dan berinteraksi langsung dengan pengguna fungsional yang terletak pada *user layer*.

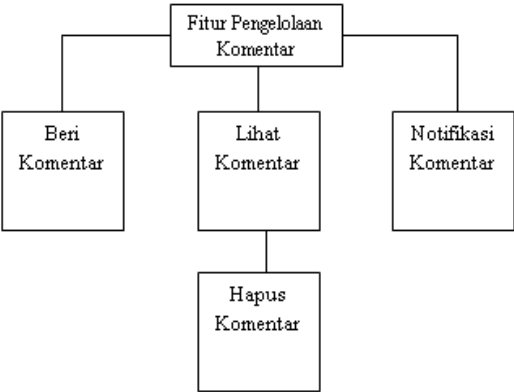
Sedangkan berikut merupakan pemetaan dekomposisi dari masing – masing *sytem layer* yang terdapat pada WikiBudaya. Pemetaan dekomposisi ini menjelaskan tentang proses fungsional apa saja yang terdapat pada masing – masing kebutuhan fungsional dari Wikibudaya. Berikut merupakan pemetaan dekomposisinya:



Gambar 4.5 Dekomposisi KF-01



Gambar 4.6 Dekomposisi KF-02



Gambar 4.7 Dekomposisi KF-03

4.5 Penentuan Software Context Model

Setelah ketiga proses untuk pencarian *Functional User Layer* dilakukan dan didapatkan *Functional User Layer* dari WikiBudaya, maka sekarang akan dilakukan proses penentuan *output* dari tahap ini, yaitu *Software Context Model*. Berikut merupakan hasil penentuan *Software Context Model* :

Tabel 4.5 Software Context Model

Tujuan Pengukuran	Mengidentifikasi kecacatan yang terdapat pada kebutuhan fungsional WikiBudaya, dengan tujuan untuk meningkatkan kualitas kebutuhan fungsional WikiBudaya.
Ruang Lingkup Pengukuran	Terbatas pada kebutuhan fungsional WikiBudaya tanpa adanya pengukuran lebih lanjut dari <i>cosmic functional point</i> .
Functional User	Admin Sistem, bertugas menjaga keamanan dan mengelola sistem dari WikiBudaya.

	Admin Reviewer, bertugas sebagai koordinator dari reviewer.
	Reviewer, bertugas melakukan review dan validasi artikel budaya.
	Kontributor, berkontribusi mengirimkan artikel - artikel budaya.
	User-Umum, masyarakat umum yang dapat mengakses WikiBudaya tanpa perlu login.

4.6 Mapping Phase

Pada tahap ini akan dilakukan penjabaran dari proses fungsional yang terdapat pada WikiBudaya, penjabaran ini dilakukan agar diketahui *sub-process* yang terkandung pada proses fungsional tersebut. Berikut merupakan daftar proses fungsional yang akan dijabarkan :

Tabel 4.6 Penjabaran Proses Fungsional

Kode Fungsional	Proses Fungsional	Kode
KF-01	Daftar Akun	UC-01.01
	Masuk Akun	UC-01.02
	Keluar Akun	UC-01.03
	Lihat Detail Akun Pribadi	UC-01.04
	Ubah Detail Akun Pribadi	UC-01.05
	Lihat Data User	UC-01.06
	Hapus Akun	UC-01.07
	Cari User	UC-01.08
	Validasi Kontributor	UC-01.09
	Validasi Reviewer	UC-01.10

	Ubah Kata Kunci	UC-01.11
KF-02	Lihat Data Master Artikel Budaya	UC-02.01
	Tambah Artikel Budaya	UC-02.02
	Ubah Artikel budaya	UC-02.03
	Hapus Artikel budaya	UC-02.04
	Cari Artikel budaya	UC-02.05
	Unggah File (lagu,video)	UC-02.06
	Unduh File (lagu,video)	UC-02.07
	Review Artikel	UC-02.08
	Validasi Artikel	UC-02.09
	Lihat Artikel	UC-02.10
	Lihat Status	UC-02.11
	Lihat Histori Artikel	UC-02.12
KF-03	Beri Komentar	UC-03.01
	Lihat Komentar	UC-03.02
	Hapus Komentar	UC-03.03
	Notifikasi Komentar	UC-03.04

Setelah didapatkan daftar proses fungsional apa saja yang terdapat pada masing - masing kebutuhan fungsional WikiBudaya, maka berikut akan ditampilkan penjabaran proses fungsional tersebut agar teridentifikasi *sub-process* apa saja yang terdapat pada tiap – tiap proses fungsional tersebut yang dibuat dalam bentuk *Generic Software Model* (untuk lebih lengkapnya dapat dilihat pada bagian Lampiran A) :

UC-01.01

Tabel 4.7 UC-01.01 Daftar Akun

Proses Fungsional	Pergerakan Data
Daftar Akun	1. Aktor memilih tombol “Menu”
	2. Sistem menampilkan pilihan “Daftar” dan “Masuk”
	3. Aktor memilih pilihan “Daftar”
	4. Sistem menampilkan <i>form</i> daftar akun (step 1) :
	a) Aktor mengisi <i>form</i> data autentifikasi yang berisikan “Nama Akun” , “Kata Kunci” , “Konfirmasi Kata Kunci”, “Peran”
	b) Apabila nama akun telah digunakan maka sistem akan menjalankan skenario alternatif I No.1
	c) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menjalankan alternatif I No.2
	d) Apabila aktor telah mengisi semua <i>field</i> yang tersedia dan tidak muncul peringatan, aktor melanjutkan skenario 3
	5. Aktor menekan tombol :
	a) “Lanjut” untuk melanjutkan ke skenario berikutnya

b) “Batal” untuk kembali ke halaman utama
6. Apabila aktor memilih peran sebagai “kontributor”, maka sistem akan menjalankan skenario No.7, dan apabila aktor memilih peran sebagai “reviewer”, maka sistem akan menjalankan skenario alternatif II.
7. Sistem menampilkan halaman data profil (step 2) :
a) Aktor mengisikan <i>formdata</i> profil, yang berisikan “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”pendidikan”, ”pekerjaan” dan “alamat email”
b) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menampilkan alternatif I No.2
8. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
a) “Ok”, maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor.
b) “Kembali”, maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi).
Alternatif I
1. Sistem akan menampilkan peringatan (notification), yang berisi bahwa “nama akun telah terdaftar”
2. Sistem akan menampilkan pesan dibawah field

	yang kosong “Kolom harus diisi.”
	Alternatif II
	1. Sistem menampilkan halaman data profil (step 2)
	a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “pendidikan”, “pekerjaan”, “alamat email” dan “berkas pendukung”
	b) Apabila terdapat field yang kosong, maka sistem akan menampilkan alternatif I No.2
	2. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
	a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor
	b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi)

UC-01.02

Tabel 4.8 UC-01.02 Masuk Akun

Proses Fungsional	Pergerakan Data
Masuk Akun	1. Aktor memasukkan “nama akun” dan “kata kunci” pada <i>form</i> masuk
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem

	3. Sistem akan memeriksa apakah <i>form</i> “nama akun” dan “kata kunci” telah sesuai dengan “data autentifikasi aktor”, jika tidak sesuai maka akan dijalankan skenario alternatif I No.1
	Alternatif I
	1. Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”

UC-01.03**Tabel 4.9 UC-01.03 Keluar Akun**

Proses Fungsional	Pergerakan Data
Keluar Akun	1. Aktor memilih menu “Keluar”
	2. Sistem kembali ke halaman utama WikiBudaya

UC-01.04**Tabel 4.10 UC-01.04 Lihat Detail Akun Pribadi**

Proses Fungsional	Pergerakan Data
Lihat Detail Akun Pribadi	1. Aktor memilih menu <i>profil</i>
	2. Sistem menampilkan tampilan data profil dan <i>privilage</i> -nya yang telah tersimpan di database.

UC-01.05**Tabel 4.11 UC-01.05 Ubah Detail Akun Pribadi**

Proses Fungsional	Pergerakan Data
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ <i>ubah profil</i> ” yang terdapat di bawah data akun lama
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama
	3. Aktor mengisi data baru pada kolom yang tersedia
	4. Aktor menekan tombol:
	a) “Ubah”, maka kemudian sistem akan menjalankan skenario sukses utama No.5
	b) “Batal”, maka sistem akan mengembalikan tampilan ke halaman akun pribadi
	5. Sistem melakukan validasi
	a) Benar, maka sistem akan menjalankan skenario utama No.6 dan 7
	b) Salah, maka sistem akan menjalankan skenario alternatif I No.1 dan 2
	6. Sistem akan mengubah data lama dengan data baru ke database
	7. Sistem menampilkan halaman akun

4.7 Measurement Phase

Setelah didapatkan *Generic Software Model*, maka akan dilakukan proses pengukuran *cosmic functional point* pada setiap *sub-process* dari *use case* WikiBudaya. Berikut merupakan beberapa tabel proses pengukuran *cosmic functional point*

terhadap proses fungsional dan *sub-process* WikiBudaya (untuk lebih lengkapnya dapat dilihat pada bagian Lampiran B).

UC-01.01

Tabel 4.12 Analisa Cacat UC-01.01

Proses Fungsional	Pergerakan Data	Tipe	CFP
Daftar Akun	1. Aktor memilih tombol “Menu”	E	1
	2. Sistem menampilkan pilihan “Daftar” dan “Masuk”	X	1
	3. Aktor memilih pilihan “Daftar”	E	1
	4. Sistem menampilkan <i>form</i> daftar akun (step 1) :	X	1
	a) Aktor mengisikan <i>form</i> data autentifikasi yang berisikan “Nama Akun” , “Kata Kunci” , “Konfirmasi Kata Kunci”, “Peran”	E	1
	b) Apabila nama akun telah digunakan maka sistem akan menjalankan skenario alternatif I No.1		0
	c) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menjalankan alternatif I No.2		0
	d) Apabila aktor telah mengisikan semua <i>field</i> yang tersedia dan tidak muncul peringatan, aktor melanjutkan skenario 3		0
	5. Aktor menekan tombol :		
	a) “Lanjut” untuk melanjutkan ke skenario berikutnya	E	1
	b) “Batal” untuk kembali ke halaman utama	E	1
	6. Apabila aktor memilih peran sebagai		

“kontributor”, maka sistem akan menjalankan skenario No.7, dan apabila aktor memilih peran sebagai “reviewer”, maka sistem akan menjalankan skenario alternatif II.		0
7. Sistem menampilkan halaman data profil (step 2) :	X	1
a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap” , “jenis kelamin” , ”tanggal lahir” , ”pendidikan” , ”pekerjaan” dan “alamat email”	E	1
b) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menampilkan alternatif I No.2		0
8. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :		
a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor.		0
b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi).		0
Alternatif I		
1. Sistem akan menampilkan peringatan (notification), yang berisi bahwa “nama akun telah terdaftar”	X	1
2. Sistem akan menampilkan pesan dibawah field yang kosong “Kolom harus diisi.”	X	1
Alternatif II		
1. Sistem menampilkan halaman data profil (step 2)	X	1

a) Aktor mengisikan <i>formdata</i> profil, yang berisikan “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “pendidikan”, “pekerjaan”, “alamat email” dan “berkas pendukung”	E	1
b) Apabila terdapat field yang kosong, maka sistem akan menampilkan alternatif I No.2		0
2. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :		
a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor		0
b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi)		0

UC-01.02

Tabel 4.13 Analisa Cacat UC-01.02

Proses Fungsional	Pergerakan Data	Tipe	CFP
Masuk Akun	1. Aktor memasukkan “nama akun” dan “kata kunci” pada <i>form</i> masuk	E	1
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem	E	1
	3. Sistem akan memeriksa apakah <i>form</i> “nama akun” dan “kata kunci” telah sesuai dengan “data autentifikasi aktor”, jika tidak sesuai maka akan dijalankan skenario alternatif I No.1		0
	Alternatif I		
	1. Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun	X	1

	tidak sesuai”		
--	---------------	--	--

UC-01.03**Tabel 4.14 Analisa Cacat UC-01.03**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Keluar Akun	1. Aktor memilih menu “Keluar”	E	1
	2. Sistem kembali ke halaman utama WikiBudaya	X	1

UC-01.04**Tabel 4.15 Analisa Cacat UC-01.04**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Detail Akun Pribadi	1. Aktor memilih menu <i>profil</i>	E	1
	2. Sistem menampilkan tampilan data profil dan <i>privilage</i> -nya yang telah tersimpan di database.	X	1

Setelah dilakukan proses pengukuran *cosmic functional point* pada setiap *sub-process* dari *use case* situs WikiBudaya, maka akan ditentukan dan dianalisa tipe kecacatan dari *use case* WikiBudaya yang telah diukur nilai *cosmic functional point*-nya. Penentuan tipe kecacatan tersebut akan disesuaikan dengan pemetaan klasifikasi kecacatan berdasarkan dokumen [13] , berikut merupakan hasil analisa kecacatannya :

Tabel 4.16 Tabel Hasil Analisa Kecacatan Use Case

Kode Use Case	Jumlah Pergerakan Data	Total <i>cfp</i>	Tipe Kecacatan (berdasarkan dokumen [13])	Catatan
UC-01.01	24	13	<i>Ambiguous or</i>	<ul style="list-style-type: none"> Terdapat informasi yang mengandung lebih dari

			<i>Unclear</i>	<p>satu interpretasi.</p> <ul style="list-style-type: none"> • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-01.02	4	3	<i>Ambiguous or Unclear</i>	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-01.03	2	2	-	-
UC-01.04	2	2	-	-
UC-01.05	9	4	<i>Ambiguous or Unclear</i>	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan. • Memiliki tipe pergerakan W, tetapi tidak memiliki tipe pergerakan R.
UC-01.06	2	2	-	-
UC-01.07	8	6	<i>Ambiguous or Unclear</i>	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan. • Memiliki tipe pergerakan W, tetapi

				tidak memiliki tipe pergerakan R.
UC-01.08	2	2	-	-
UC-01.09	2	2	<i>Incorrect information</i>	Informasi dari seluruh pergerakan data yang diberikan oleh <i>use case</i> ini tidak sesuai dengan judul <i>use case</i> yang dimaksud.
UC-01.10	2	2	<i>Incorrect information</i>	Informasi dari seluruh pergerakan data yang diberikan oleh <i>use case</i> ini tidak sesuai dengan judul <i>use case</i> yang dimaksud.
UC-01.11	3	3	<i>Missing or Incomplete</i>	Memiliki pergerakan data yang bagus tetapi terdapat informasi yang tidak disebutkan pada <i>use case</i> ini, yang menyebabkan informasi dari <i>use case</i> ini tidaklah lengkap dan tidak sesuai dengan artifak.
UC-02.01	2	2	-	-
UC-02.02	2	1	<i>Incorrect information</i>	Informasi dari seluruh pergerakan data yang diberikan oleh <i>use case</i> ini tidak sesuai dengan judul <i>use case</i> yang dimaksud.
UC-02.03	6	3	<i>Ambiguous or Unclear</i>	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-02.04	8	6	<i>Ambiguous or</i>	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari

			<i>Unclear</i>	<p>satu interpretasi.</p> <ul style="list-style-type: none"> • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-02.05	2	2	<i>Missing or Incomplete</i>	<p>Terdapat informasi yang tidak disebutkan pada <i>use case</i> ini, yang menyebabkan informasi dari <i>use case</i> ini tidaklah lengkap dan tidak sesuai dengan artifak.</p>
UC-02.06	8	4	Ambiguous or Unclear	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-02.07	3	3	-	-
UC-02.08	2	2	<i>Redundant or Duplicate</i>	<p>Memiliki pergerakan data yang bagus tetapi informasi pada <i>use case</i> ini merupakan duplikasi dari kebutuhan lain atau sebagian dari informasi pada <i>use case</i> ini sudah dijelaskan pada <i>use case</i> yang lain.</p>
UC-02.09	7	5	<i>Redundant or Duplicate & Ambiguous or Unclear</i>	<ul style="list-style-type: none"> • Memiliki pergerakan data yang bagus tetapi informasi pada <i>use case</i> ini merupakan duplikasi dari kebutuhan lain atau sebagian dari informasi pada <i>use case</i> ini sudah dijelaskan pada <i>use case</i> yang lain. • Terdapat beberapa

				pergerakan data yang memiliki dua proses dalam satu pergerakan.
UC-02.10	4	4	-	-
UC-02.11	3	3	-	-
UC-02.12	1	1	<i>Missing or Incomplete</i>	<ul style="list-style-type: none"> • Terdapat informasi yang tidak disebutkan pada <i>use case</i> ini, yang menyebabkan informasi dari <i>use case</i> ini tidaklah lengkap dan tidak sesuai dengan artifak. • Memiliki tipe pergerakan X, tetapi tidak memiliki tipe pergerakan E.
UC-03.01	3	2	<i>Missing or Incomplete</i>	<ul style="list-style-type: none"> • Terdapat informasi yang tidak disebutkan pada <i>use case</i> ini, yang menyebabkan informasi dari <i>use case</i> ini tidaklah lengkap dan tidak sesuai dengan artifak. • Memiliki tipe pergerakan W, tetapi tidak memiliki tipe pergerakan R.
UC-03.02	2	2	-	-
UC-03.03	7	4	Ambiguous or Unclear	<ul style="list-style-type: none"> • Terdapat informasi yang mengandung lebih dari satu interpretasi. • Terdapat beberapa pergerakan data yang memiliki dua proses dalam satu pergerakan. • Memiliki tipe pergerakan W, tetapi

				tidak memiliki tipe pergerakan R.
UC-03.04	3	3	-	-

Dari hasil analisa kecacatan pada 27 *use case* WikiBudaya, didapatkan 17 *use case* memiliki kecacatan. Dari 17 *use case* tersebut, 4 diantaranya tergolong pada tipe *missing or incomplete*, sedangkan 3 diantaranya tergolong tipe *incorrect information*, lalu 2 diantaranya tergolong pada tipe *redundant or duplicate* dan 8 lainnya merupakan sebuah *use case* yang memiliki tipe kecacatan *ambiguous or unclear*. Maka dari itu, untuk tahap selanjutnya akan dilakukan proses *refactoring* untuk memperbaiki *use case* yang bermasalah.

Sedangkan sesuai dengan konsep pemahaman kualitas yang telah dijelaskan pada bab sebelumnya, kualitas dari *use case* WikiBudaya masih tidak sesuai dengan persyaratan kebutuhan dan belum bisa memenuhi kebutuhan pelanggannya. Hal tersebut dapat dilihat dari penilaian total *cfp* pada masing – masing *use case* WikiBudaya, masih terdapat banyak *use case* yang tidak memiliki total nilai *cfp* yang sesuai dengan jumlah pergerakan data yang diharapkan. Berikut merupakan tabel yang berisi informasi mengenai total *cfp* dari masing – masing *use case* WikiBudaya :

Tabel 4.17 Tabel Presentase Kesesuaian Use Case

Kode Use Case	Jumlah Pergerakan Data	Total <i>cfp</i>	Presentase Kesesuaian Pergerakan Data
UC-01.01	24	13	54%
UC-01.02	4	3	75%
UC-01.03	2	2	100%
UC-01.04	2	2	100%

UC-01.05	9	4	44%
UC-01.06	2	2	100%
UC-01.07	8	6	75%
UC-01.08	2	2	100%
UC-01.09	2	2	100%
UC-01.10	2	2	100%
UC-01.11	3	3	100%
UC-02.01	2	2	100%
UC-02.02	2	1	50%
UC-02.03	6	3	50%
UC-02.04	8	6	75%
UC-02.05	2	2	100%
UC-02.06	8	4	50%
UC-02.07	3	3	100%
UC-02.08	2	2	100%
UC-02.09	7	5	71%
UC-02.10	4	4	100%
UC-02.11	3	3	100%
UC-02.12	1	1	100%
UC-03.01	3	2	67%
UC-03.02	2	2	100%
UC-03.03	7	4	57%
UC-03.04	3	3	100%

Sesuai dengan Tabel 4.17 , diketahui masih terdapat 11 *use case* WikiBudaya yang tidak memenuhi kesesuaian persyaratan kebutuhan. Maka dari itu, *use case* yang terdapat pada WikiBudaya masih belum dapat dikatakan berkualitas karena masih terdapat beberapa *use case* yang tidak sesuai dengan konsep kualitas, yaitu kesesuaian dengan persyaratan kebutuhan dan

pemenuhan kebutuhan pelanggan, maka dari itu akan dilakukan suatu perbaikan terhadap *use case* yang bermasalah tersebut.

4.8 Proses Refactoring

Dalam proses ini akan dilakukan perbaikan terhadap *use case* yang diketahui memiliki kecacatan, tetapi sebelum dilakukan perbaikan, *use case* yang memiliki kecacatan tersebut akan dianalisa lagi mengenai peluang dan solusi *refactoring* nya. Berikut merupakan tabel analisa peluang dan solusi *refactoring* terhadap *use case* yang memiliki kecacatan :

Tabel 4.18 Tabel identifikasi peluang & solusi Refactoring

Kode <i>Use Case</i>	Tipe Kecacatan (berdasarkan dokumen [13])	<i>Refactoring Opportunity</i>	Alasan	<i>Refactoring</i>
UC-01.01	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract requirements & extract alternative flows
UC-01.02	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows
UC-01.05	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows

UC-01.07	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows
UC-01.09	<i>Incorrect information</i>	<i>Lazy Requirement</i>	Informasi yang disajikan memberikan dampak yang tidak jelas atau tidak sesuai.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu menyesuaikan dengan artefak yang ada.
UC-01.10	<i>Incorrect information</i>	<i>Lazy Requirement</i>	Informasi yang disajikan memberikan dampak yang tidak jelas atau tidak sesuai.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu menyesuaikan dengan artefak yang ada.
UC-01.11	<i>Missing or Incomplete</i>	<i>Lazy Requirement</i>	Terdapat informasi yang tidak disajikan, sehingga membuat <i>use case</i> ini tidak lengkap.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu melengkapi informasi yang kurang lengkap saja.
UC-02.02	<i>Incorrect information</i>	<i>Lazy Requirement</i>	Informasi yang disajikan memberikan dampak yang tidak jelas atau	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu menyesuaikan

			tidak sesuai.	kan dengan artefak yang ada.
UC-02.03	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows
UC-02.04	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows
UC-02.05	<i>Missing or Incomplete</i>	<i>Lazy Requirement</i>	Terdapat informasi yang tidak disajikan, sehingga membuat <i>use case</i> ini tidak lengkap.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu melengkapi informasi yang kurang lengkap saja.
UC-02.06	<i>Ambiguous or Unclear</i>	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.	Extract alternative flows
UC-02.08	<i>Redundant or Duplicate</i>	<i>Duplicate Activities</i>	Terdapat pengulangan informasi atau kebutuhan yang sudah terdapat pada kebutuhan	Move Activity

			yang lainnya	
UC-02.09	<i>Redundant or Duplicate & Ambiguous or Unclear</i>	<i>Duplicate Activities & Large Requirement</i>	<p>Terdapat pengulangan informasi atau kebutuhan yang sudah terdapat pada kebutuhan yang lainnya</p> <p>Terdapat <i>main flow</i> yang mengandung beberapa proses dalam satu waktu.</p>	Extract alternative flows
UC-02.12	<i>Missing or Incomplete</i>	<i>Lazy Requirement</i>	Terdapat informasi yang tidak disajikan, sehingga membuat <i>use case</i> ini tidak lengkap.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu melengkapi informasi yang kurang lengkap saja.
UC-03.01	<i>Missing or Incomplete</i>	<i>Lazy Requirement</i>	Terdapat informasi yang tidak disajikan, sehingga membuat <i>use case</i> ini tidak lengkap.	Tidak perlu dilakukan <i>refactoring</i> , hanya perlu melengkapi informasi yang kurang lengkap saja.
UC-03.03	Ambiguous or Unclear	<i>Large Requirement</i>	Terdapat <i>main flow</i> yang mengandung beberapa proses	Extract alternative flows

			dalam satu waktu.	
--	--	--	-------------------	--

Berikut merupakan tahap *refactoring* terhadap *use case* WikiBudaya yang memiliki kecacatan:

- **UC-01.01**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Ectract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.19 UC-01.01 sebelum refactoring

Proses Fungsional	Pergerakan Data
Daftar Akun	1. Aktor memilih tombol “Menu”
	2. Sistem menampilkan pilihan “Daftar” dan “Masuk”
	3. Aktor memilih pilihan “Daftar”
	4. Sistem menampilkan <i>form</i> daftar akun (step 1) :
	a) Aktor mengisi <i>form</i> data autentifikasi yang berisikan “Nama Akun” , “Kata Kunci” , “Konfirmasi Kata Kunci”, “Peran”
	b) Apabila nama akun telah digunakan maka sistem akan menjalankan skenario alternatif I No.1

c) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menjalankan alternatif I No.2
d) Apabila aktor telah mengisi semua <i>field</i> yang tersedia dan tidak muncul peringatan, aktor melanjutkan skenario 3
5. Aktor menekan tombol :
a) “Lanjut” untuk melanjutkan ke skenario berikutnya
b) “Batal” untuk kembali ke halaman utama
6. Apabila aktor memilih peran sebagai “kontributor”, maka sistem akan menjalankan skenario No.7, dan apabila aktor memilih peran sebagai “reviewer”, maka sistem akan menjalankan skenario alternatif II.
7. Sistem menampilkan halaman data profil (step 2) :
a) Aktor mengisi <i>form</i> data profil, yang berisikan “nama lengkap” , “jenis kelamin” , ”tanggal lahir” , ”pendidikan” , ”pekerjaan” dan “alamat email”
b) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menampilkan alternatif I No.2
8. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor.
b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi).
Alternatif I

	1. Sistem akan menampilkan peringatan (notification), yang berisi bahwa “nama akun telah terdaftar”
	2. Sistem akan menampilkan pesan dibawah field yang kosong “Kolom harus diisi.”
	Alternatif II
	1. Sistem menampilkan halaman data profil (step 2)
	a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “pendidikan”, “pekerjaan”, “alamat email” dan “berkas pendukung”
	b) Apabila terdapat field yang kosong, maka sistem akan menampilkan alternatif I No.2
	2. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
	a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor
	b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi)

Tabel 4.20 UC-01.01 setelah refactoring

Proses Fungsional	Pergerakan Data
Daftar Akun	1. Aktor memilih tombol “Menu”
	2. Sistem menampilkan pilihan “Daftar” dan “Masuk”
	3. Aktor memilih pilihan “Daftar”
	4. Sistem menampilkan <i>form</i> “Daftar Akun”

5. Aktor mengisi <i>form</i> data profil, yang berisikan “nama akun” , ”kata kunci” , ”konfirmasi kata kunci” dan “peran”
6. Aktor memilih peran sebagai kontributor atau <i>reviewer</i>
7. Aktor menekan tombol “Lanjut”
8. Sistem memvalidasi dan memproses data akun pada database kontributor / <i>reviewer</i>
9. Sistem membaca hasil vaildasi data akun pada database kontributor / <i>reviewer</i>
10. Sistem menampilkan halaman <i>form</i> “Data Profil”
11. Aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap” , “jenis kelamin” , ”tanggal lahir” , ”pendidikan” , ”pekerjaan” dan “alamat email”
12. Aktor menekan tombol “Selesai”
13. Sistem memvalidasi dan memproses data profil pada database kontributor / <i>reviewer</i>
14. Sistem membaca hasil vaildasi data profil pada database
15. Sistem menampilkan halaman utama kontributor / <i>reviewer</i>
Alternatif
Jika memilih sebagai <i>reviewer</i> , maka aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap”, “jenis kelamin”, ”tanggal lahir” , ”bidang studi” , “pendidikan”, ”pekerjaan” , “alamat email” dan “berkas penunjang”
Exception
9.a Sistem membaca hasil vaildasi data akun pada database kontributor / <i>reviewer</i> dan ternyata ada data akun yang tidak valid
9.b Sistem kembali menampilkan <i>form</i> “Daftar Akun”
9.c Sistem akan menampilkan peringatan, yang berisi

	bahwa “Nama Akun Telah Terdaftar” atau “Semua Kolom Harus Diisi”
	14.a Sistem membaca hasil validasi data profil pada database kontributor / <i>reviewer</i> dan ternyata ada data profil yang tidak valid
	14.b Sistem kembali menampilkan <i>form</i> “Data Profil”
	14.c Sistem akan menampilkan peringatan, yang berisi bahwa “Semua Kolom Harus Diisi”

- **UC-01.02**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Extract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.21 UC-01.02 sebelum refactoring

Proses Fungsional	Pergerakan Data
Masuk Akun	1. Aktor memasukkan “nama akun” dan “kata kunci” pada <i>form</i> masuk
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam system
	3. Sistem akan memeriksa apakah <i>form</i> “nama akun” dan “kata kunci” telah sesuai dengan “data autentifikasi aktor”, jika tidak sesuai maka akan dijalankan skenario alternatif I No.1
	Alternatif I

	1. Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”
--	---

Tabel 4.22 UC-01.02 setelah refactoring

Proses Fungsional	Pergerakan Data
Masuk Akun	1. Aktor memasukkan “Nama Akun” dan “Kata Kunci” pada <i>form</i> masuk
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam system
	3. Sistem mengecek dan memvalidasi data akun dan kata kunci pada database yang ada pada sistem
	4. Sistem membaca hasil vaildasi data akun dan kata kunci pada database yang ada pada sistem data autentikasi aktor
	5. Sistem menmpilkan halaman utama aktor
	Exception
	4.a Sistem membaca hasil vaildasi data akun dan kata kunci pada database yang ada pada sistem dan ternyata data akun tersebut tidak valid
	4.b Sistem kembali ke halaman login
	4.c Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”

- **UC-01.05**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Ectract Alternative Flow
 Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.23 UC-01.05 sebelum refactoring

Proses Fungsional	Pergerakan Data
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ <i>ubah profil</i> ” yang terdapat di bawah data akun lama
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama
	3. Aktor mengisi data baru pada kolom yang tersedia
	4. Aktor menekan tombol :
	a) “Ubah”, maka kemudian sistem akan menjalankan skenario sukses utama No.5
	b) “Batal”, maka sistem akan mengembalikan tampilan ke halaman akun pribadi
	5. Sistem melakukan validasi
	a) Benar, maka sistem akan menjalankan skenario utama No.6 dan 7
	b) Salah, maka sistem akan menjalankan skenario alternatif I No.1 dan 2
	6. Sistem akan mengubah data lama dengan data baru ke database
	7. Sistem menampilkan halaman akun pribadi

Tabel 4.24 UC-01.05 setelah refactoring

Proses Fungsional	Pergerakan Data
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ubah profil” yang terdapat di bawah data akun lama
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama
	3. Aktor mengisi data baru pada kolom yang tersedia
	4. Aktor menekan tombol “Ubah”
	5. Sistem memvalidasi dan mengubah data akun pribadi terbaru pada database yang ada pada sistem
	6. Sistem membaca hasil validasi data akun pribadi terbaru pada database tersebut
	7. Sistem menampilkan” Halaman Akun Pribadi”
	Exception
	4.a Jika aktor menekan “Batal”
	4.b Sistem menampilkan “Halaman Akun Pribadi”
	6.a Sistem membaca hasil validasi data akun pribadi terbaru pada database tersebut dan ternyata data akun pribadi tersebut tidak valid
	6.b Sistem menampilkan notifikasi bahwa data akun pribadi salah.

- **UC-01.07**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Extract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut

menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.25 UC-01.07 sebelum refactoring

Proses Fungsional	Pergerakan Data
Hapus Akun	1. Aktor memilih data akun yang akan dihapus
	2. Aktor menekan tombol “Hapus” yang terletak disamping nama akun
	3. Sistem akan menampilkan <i>pop up window</i> berupa pertanyaan “Apakah anda yakin untuk menghapus akun ini?”
	4. Aktor menekan tombol :
	a) “Ok”, maka sistem akan menjalankan skenario utama No.4, 5, 6
	b) “Cancel”, maka sistem akan kembali menampilkan halaman daftar user
	5. Sistem akan menghapus akun dari database
	6. Sistem kembali ke halaman daftar akun
	7. Sistem menampilkan notifikasi “akun telah berhasil di hapus

Tabel 4.26 UC-01.07 setelah refactoring

Proses Fungsional	Pergerakan Data
Hapus Akun	1. Aktor memilih data akun yang akan dihapus
	2. Aktor menekan tombol “Hapus” yang terletak disamping nama akun
	3. Sistem akan menampilkan <i>pop up window</i> berupa

pertanyaan “Apakah anda yakin untuk menghapus akun ini?”
4. Aktor menekan tombol “Ok”
5. Sistem akan menghapus akun dari database yang ada pada sistem
6. Sistem akan membaca hasil pembaharuan dari database tersebut
7. Sistem kembali ke halaman “daftar akun”
8. Sistem menampilkan notifikasi “akun telah berhasil di hapus”
Exception
4.a Jika aktor menekan “Batal”
4.b Sistem kembali ke halaman “daftar akun”

- **UC-01.09**

Permasalahan : Informasi yang disajikan memberikan dampak yang tidak jelas / tidak sesuai.

Tipe : Incorrect Information

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Merubah informasi yang disajikan agar sesuai dengan artefak.

Hasil :

Tabel 4.27 UC-01.09 sebelum refactoring

Proses Fungsional	Pergerakan Data
Validasi Kontributor	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian
	2. Sistem menampilkan hasil pencarian berdasarkan

	kata kunci yang dimasukkan, dari semua peran akun
--	---

Tabel 4.28 UC-01.09 setelah refactoring

Proses Fungsional	Pergerakan Data
Validasi Kontributor	1. Sistem menampilkan pesan ada pengguna baru
	2. Aktor membuka detail pengguna
	3. Aktor menekan tombol “Validasi”
	4. Sistem kembali ke halaman utama admin sistem

- **UC-01.10**

Permasalahan : Informasi yang disajikan memberikan dampak yang tidak jelas / tidak sesuai.

Tipe : Incorrect Information

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Merubah informasi yang disajikan agar sesuai dengan artefak.

Hasil :

Tabel 4.29 UC-01.10 sebelum refactoring

Proses Fungsional	Pergerakan Data
Validasi Reviewer	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun

Tabel 4.30 UC-01.10 setelah refactoring

Proses Fungsional	Pergerakan Data
Validasi Reviewer	1. Sistem menampilkan pesan ada <i>reviewer</i> baru
	2. Aktor membuka detail <i>reviewer</i>
	3. Aktor menekan tombol “Validasi”
	4. Sistem kembali ke halaman utama admin <i>reviewer</i>

- **UC-01.11**

Permasalahan : Terdapat informasi yang tidak disajikan, sehingga membuat use case ini tidak lengkap.

Tipe : Missing or Incomplete

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Melengkapi informasi yang kurang lengkap saja.

Hasil :

Tabel 4.31 UC-01.11 sebelum refactoring

Proses Fungsional	Pergerakan Data
Ubah Kata Kunci	1. Sistem menampilkan halaman ubah profil
	2. Aktor menekan tombol ubah kata kunci
	3. Sistem menampilkan halaman form ganti kata kunci

Tabel 4.32 UC-01.11 setelah refactoring

Proses	Pergerakan Data
--------	-----------------

Fungsional	
Ubah Kata Kunci	1. Sistem menampilkan halaman dari profil pengguna
	2. Aktor menekan tombol ubah kata kunci
	3. Sistem menampilkan halaman untuk ubah kata kunci
	4. Aktor mengisi field untuk mengubah kata kunci
	5. Aktor menekan tombol “Simpan”
	6. Sistem memvalidasi dan mengubah data kata kunci terbaru pada database yang ada pada sistem
	7. Sistem membaca hasil validasi data kata kunci terbaru pada database tersebut
	8. Sistem kembali menampilkan halaman dari profil pengguna
	Exception
	7.a Sistem membaca hasil validasi data kata kunci terbaru pada database tersebut dan ternyata data kata kunci tersebut tidak valid
	7.b Sistem menampilkan notifikasi bahwa kata kunci salah.

- **UC-02.02**

Permasalahan : Informasi yang disajikan memberikan dampak yang tidak jelas / tidak sesuai.

Tipe : Incorrect Information

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Merubah informasi yang disajikan agar sesuai dengan artefak.

Hasil :

Tabel 4.33 UC-02.02 sebelum refactoring

Proses Fungsional	Pergerakan Data
--------------------------	------------------------

Tambah Artikel Budaya	1. Aktor menekan tombol “validate”
	2. Sistem menyimpan artikel budaya yang telah di validasi

Tabel 4.34 UC-02.02 setelah refactoring

Proses Fungsional	Pergerakan Data
Tambah Artikel Budaya	1. Aktor menekan <i>tab menu</i> “Tulis Artikel”
	2. Sistem akan menampilkan halaman untuk membuat artikel baru
	3. Aktor menuliskan artikel baru pada <i>field</i> yang tersedia
	4. Aktor menekan tombol “Simpan”
	5. Sistem menyimpan artikel baru dari aktor pada database
	6. Sistem membaca artikel baru dari aktor yang telah disimpan pada database
	7. Sistem menampilkan halaman “Daftar Master artikel”
	Exception
	4.a Jika aktor menekan “Batal”
	4.b Sistem menampilkan halaman “Daftar Master artikel”

- **UC-02.03**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Extract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.35 UC-02.03 sebelum refactoring

Proses Fungsional	Pergerakan Data
Ubah Artikel	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah
	2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah terisi <i>value</i> data lama
	3. Aktor mengisi data baru pada artikel
	4. Aktor menekan tombol :
	a) “simpan”, maka sistem akan menyimpan perubahan dan menjalankan skenario 5
	b) “batal”, maka sistem akan mengembalikan tampilan ke halaman artikel saya
	5. Sistem akan mengirimkan artikel baru ke reviewer

Tabel 4.36 UC-02.03 setelah refactoring

Proses Fungsional	Pergerakan Data
Ubah Artikel	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah
	2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah terisi <i>value</i> data lama
	3. Aktor mengisi data baru pada artikel
	4. Aktor menekan tombol “simpan”
	5. Sistem akan mengirimkan artikel baru ke database artikel

	6. Sistem membaca data artikel yang telah diperbaharui pada database artikel
	7. Sistem kembali ke halaman “artikel saya”
	Exception
	4.a Jika aktor menekan “Batal”
	4.b Sistem kembali ke halaman “artikel saya” Artikel]

• **UC-02.04**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Extract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.37 UC-02.04 sebelum refactoring

Proses Fungsional	Pergerakan Data
Hapus Artikel Budaya	1. Aktor memilih artikel yang akan dihapus
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus
	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya?”
	4. Aktor menekan tombol:
	a) “Ok”, maka sistem akan menjalankan skenario utama nomor 4,5,6

	b) “Batal”, maka sistem akan kembali menampilkan halaman daftar artikel
	5. Sistem akan menghapus artikel dari database
	6. Sistem kembali ke halaman “daftar artikel”
	7. Sistem menampilkan notifikasi “artikel berhasil dihapus”

Tabel 4.38 UC-02.04 setelah refactoring

Proses Fungsional	Pergerakan Data
Hapus Artikel Budaya	1. Aktor memilih artikel yang akan dihapus
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus
	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya?”
	4. Aktor menekan “Ok”
	5. Sistem akan menghapus artikel dari database artikel
	6. Sistem membaca database artikel yang telah diperbaharui
	7. Sistem kembali ke halaman “daftar artikel”
	8. Sistem menampilkan notifikasi “artikel berhasil dihapus”
	Exception
	4.a Jika aktor menekan “Batal”
	4.b Sistem kembali ke halaman “daftar artikel”

- UC-02.05

Permasalahan : Terdapat informasi yang tidak disajikan, sehingga membuat use case ini tidak lengkap.

Tipe : Missing or Incomplete

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Melengkapi informasi yang kurang lengkap saja.

Hasil :

Tabel 4.39 UC-02.05 sebelum refactoring

Proses Fungsional	Pergerakan Data
Cari Artikel Budaya	1. Aktor memasukkan kata / <i>keyword</i> pencarian pada kolom pencarian
	2. Begitu <i>keyword</i> diketikkan, maka sistem akan menampilkan hasil pencarian

Tabel 4.40 UC-02.05 setelah refactoring

Proses Fungsional	Pergerakan Data
Cari Artikel Budaya	1. Aktor memasukkan kata / <i>keyword</i> pencarian pada kolom pencarian
	2. Aktor menekan tombol “Cari Budaya”
	3. Maka sistem akan menampilkan hasil pencarian

- **UC-02.06**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Ectract Alternative Flow
 Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.
 Hasil :

Tabel 4.41 UC-02.06 sebelum refactoring

Proses Fungsional	Pergerakan Data
Unggah File	1. Aktor memilih tombol “ Unggah file budaya”
	2. Sistem menampilkan halaman “pop-up” , yang berisikan “kategori budaya”, “judul file budaya”, dan “tombol browse”
	3. Untuk upload file, ada ketentuan ukuran dan format file yang akan di upload. Untuk ukuran maksimal berukuran 5 Mb, dan format yang digunakan haruslah berformat : mp3/mp4, avi. Apabila tidak sesuai maka sistem akan menampilkan alternatif I.no 1, dan apabila format yang digunakan tidak sesuai maka file tidak akan muncul pada file yang di browser
	4. Setelah semua <i>field</i> terisi, dan ukuran file tidak melebihi ukuran yang ditentukan Aktor dapat menekan tombol :
	a) “Ok” ,maka sistem akan mengunggah file budaya ke reviewer dan menjalankan skenario 5
	b) “Batal” untuk membatalkan pengunggahan file budaya
	5. Sistem kembali ke halaman “tuliskan artikel”

	Alternatif I
	1. Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal

Tabel 4.42 UC-02.06 setelah refactoring

Proses Fungsional	Pergerakan Data
Unggah File	1. Aktor memilih tombol “ Unggah file budaya”
	2. Sistem menampilkan halaman “Upload”
	3. Aktor meng- <i>upload file</i> sesuai dengan ketentuan ukuran dan format <i>file</i>
	4. Aktor menekan “upload”
	5. Sistem mengunggah file tersebut ke dalam database artikel
	6. Sistem membaca file yang telah diunggah pada database artikel
	7. Sistem kembali ke halaman “tuliskan artikel”
	Exception
	3.a Aktor meng- <i>upload file</i> tidak sesuai dengan ketentuan ukuran dan format <i>file</i>
	3.b File tidak akan muncul pada file yang di browse Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal
	5.a Jika aktor menekan “Batal”
	5.b Sistem kembali ke halaman “tuliskan artikel”

- **UC -02.08 dan UC-02.09**

Permasalahan : Terdapat informasi yang merupakan duplikasi dari kebutuhan lain dan terdapat *main flow* yang mengandung beberapa proses dalam satu waktu

- Tipe : Redundant / Duplicate
- Opportunity : Duplicate Activities & Large Requirement
- Refactoring : Move Activity & Extract Alternative Flows
- Solusi : Memindahkan informasi yang kembar pada *use case* yang dituju dan mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.
- Hasil :

Tabel 4.43 UC-02.09 sebelum refactoring

Proses Fungsional	Pergerakan Data
Review Artikel	1. Aktor memilih tombol “ Validasi”
	2. Sistem menampilkan halaman artikel yang di review

Tabel 4.44 UC-02.09 setelah refactoring (Move Activity)

Proses Fungsional	Pergerakan Data
Validasi Artikel	1. Aktor memilih tombol “ Validasi”
	2. Sistem menampilkan halaman artikel yang di review
	3. Setelah melakukan review terhadap artikel budaya, Aktor menekan tombol :
	a) “Valid”, maka sistem akan menjalankan skenario utama nomor 4,5,6
	b) “Tidak disetujui”, maka sistem akan menampilkan skenario utama nomor 6

	4. Sistem akan menambahkan artikel pada database
	5. Sistem akan menampilkan artikel yang telah divalidasi tersebut
	6. Sistem menampilkan halaman “list artikel”

Tabel 4.45 UC-02.09 setelah refactoring (Extract Alternatvie Flows)

Proses Fungsional	Pergerakan Data
Validasi Artikel	1. Aktor memilih tombol “ Validasi”
	2. Sistem menampilkan halaman artikel yang di <i>review</i>
	3. Aktor menekan tombol “Setujui”
	4. Sistem akan menambahkan artikel pada database artikel
	5. Sistem membaca artikel terbaru yang ditambahkan pada daabase artikel
	6. Sistem menampilkan halaman “list artikel”
	Exception
	3.a Jika aktor menekan “Tidak disetujui”
	3.b Sistem menampilkan halaman “list artikel”

- **UC-02.12**

Permasalahan : Terdapat informasi yang tidak disajikan, sehingga membuat use case ini tidak lengkap.

Tipe : Missing or Incomplete

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Melengkapi informasi yang kurang lengkap saja.

Hasil :

Tabel 4.46 UC-02.12 sebelum refactoring

Proses Fungsional	Pergerakan Data
Lihat Histori Artikel	1. Sistem menampilkan <i>history</i> artikel yang pernah di kirimkan oleh kontributor

Tabel 4.47 UC-02.12 setelah refactoring

Proses Fungsional	Pergerakan Data
Lihat Histori Artikel	1. Aktor menekan <i>tab menu</i> “Akitiftas Histori” pada halaman utama contributor
	2. Sistem menampilkan <i>history</i> artikel yang pernah di kirimkan oleh kontributor

- **UC-03.01**

Permasalahan : Terdapat informasi yang tidak disajikan, sehingga membuat use case ini tidak lengkap.

Tipe : Missing or Incomplete

Opportunity : Lazy Requirement

Refactoring : Tidak Diperlukan

Solusi : Melengkapi informasi yang kurang lengkap saja.

Hasil :

Tabel 4.48 UC-03.01 sebelum refactoring

Proses Fungsional	Pergerakan Data
Beri	1. Aktor mengisikan komentar pada kolom “ box

Komentar	komentar”
	2. Sistem menampilkan komentar yang baru dibuat aktor
	3. Sistem menyimpan komentar dari aktor

Tabel 4.49 UC-03.01 setelah refactoring

Proses Fungsional	Pergerakan Data
Beri Komentar	1. Aktor menekan tab menu “komentar”
	2. Sistem menampilkan form komentar
	3. Aktor mengisikan komentar pada kolom “ box komentar”
	4. Aktor menekan tombol “Kirim”
	5. Sistem menyimpan komentar dari aktor pada database komentar
	6. Sistem membaca komentar dari aktor yang telah disimpan pada database komentar
	7. Sistem menampilkan komentar yang baru dibuat aktor

- **UC-03.03**

Permasalahan : Terdapat *main flow* yang mengandung beberapa proses dalam satu waktu.

Tipe :

Opportunity : Large Requirement

Refactoring : Ectract Alternative Flow

Solusi : Mengekstrak informasi yang mengandung beberapa proses tersebut menjadi *alternative flows* atau *exception*.

Hasil :

Tabel 4.50 UC-03.03 sebelum refactoring

Proses	Pergerakan Data
--------	-----------------

Fungsional	
Hapus Komentar	1. Aktor memilih menu komentar
	2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya
	3. Aktor memilih tanda “silang (hapus)” pada komentar yang akan dihapus , aktor akan menekan tombol “ X (silang)” sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini”
	4. Aktor menekan tombol :
	a) “ OK “ sistem akan menghapus komentar dan menjalankan skenario 5.6
	b) “ Batal” Sistem menjalankan skenario 6
	5. Sistem memperbaharui database komentar
	6. Sistem menampilkan halaman komentar

Tabel 4.51 C-03.03 setelah refactoring

Proses Fungsional	Pergerakan Data
Hapus Komentar	1. Aktor memilih menu komentar
	2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya
	3. Aktor menekan tanda “silang (hapus)” pada komentar yang akan dihapus
	4. Sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini”
	5. Aktor menekan “OK”
	6. Sistem memperbaharui database komentar
	7. Sistem membaca data komentar yang telah

	diperbaharui pada database komentar
	8. Sistem menampilkan halaman komentar
	Exception
	5.a Jika aktor menekan “Batal”
	5.b Sistem kembali menampilkan halaman komentar

4.9 Proses Re-Measurement Phase

Setelah didapatkan hasil refactoring dari *use case* yang memiliki kecacatan, maka selanjutnya akan dilakukan pengidentifikasian kecacatan ulang terhadap *use case* hasil *refactoring* tersebut dengan tujuan agar mengetahui apakah *use case* hasil *refactoring* tersebut masih memiliki kecacatan atau tidak. Berikut merupakan tabel pengukuran ulang kecacatan terhadap *use case* WikiBudaya yang telah di refactoring (untuk lebih lengkapnya dapat dilihat pada bagian Lampiran D) :

Tabel 4.52 Re-Measurement Phase UC Daftar Akun Kontributor

Proses Fungsional	Pergerakan Data	Tipe	CFP
Daftar Akun	1. Aktor memilih tombol “Menu”	E	1
	2. Sistem menampilkan pilihan “ Daftar” dan “Masuk”	X	1
	3. Aktor memilih pilihan “Daftar”	E	1
	4. Sistem menampilkan <i>form</i> “Daftar Akun”	X	1
	5. Aktor mengisikan <i>form</i> data profil, yang berisikan “nama akun” , ”kata kunci” , ”konfirmasi kata kunci” dan “peran”	E	1
	6. Aktor memilih peran sebagai kontributor atau <i>reviewer</i>	E	1
	7. Aktor menekan tombol “Lanjut”	E	1

8. Sistem memvalidasi dan memproses data akun pada database kontributor / <i>reviewer</i>	W	1
9. Sistem membaca hasil validasi data akun pada database kontributor / <i>reviewer</i>	R	1
10. Sistem menampilkan halaman <i>form</i> “Data Profil”	X	1
11. Aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “pendidikan”, “pekerjaan” dan “alamat email”	E	1
12. Aktor menekan tombol “Selesai”	E	1
13. Sistem memvalidasi dan memproses data profil pada database kontributor / <i>reviewer</i>	W	1
14. Sistem membaca hasil validasi data profil pada database kontributor / <i>reviewer</i>	R	1
15. Sistem menampilkan halaman utama kontributor / <i>reviewer</i>	X	1
Alternatif		
Jika memilih sebagai <i>reviewer</i> , maka aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “bidang studi”, “pendidikan”, “pekerjaan”, “alamat email” dan “berkas penunjang”	E	1
Exception		
9.a Sistem membaca hasil validasi data akun pada database kontributor / <i>reviewer</i> dan ternyata ada data akun yang tidak valid	R	1
9.b Sistem kembali menampilkan <i>form</i> “Daftar Akun”	X	1
9.c Sistem akan menampilkan peringatan,	X	1

	yang berisi bahwa “Nama Akun Telah Terdaftar” atau “Semua Kolom Harus Diisi”		
	14.a Sistem membaca hasil vaildasi data profil pada database kontributor / <i>reviewer</i> dan ternyata ada data profil yang tidak valid	R	1
	14.b Sistem kembali menampilkan <i>form</i> “Data Profil”	X	1
	14.c Sistem akan menampilkan peringatan, yang berisi bahwa “Semua Kolom Harus Diisi”	X	1

Tabel 4.53 Re-Measurement Phase UC Masuk Akun

Proses Fungsional	Pergerakan Data	Tipe	CFP
Masuk Akun	1. Aktor memasukkan “Nama Akun” dan “Kata Kunci” pada <i>form</i> masuk	E	1
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem	E	1
	3. Sistem mengecek dan memvalidasi data akun dan kata kunci pada database data autentikasi aktor	W	1
	4. Sistem membaca hasil vaildasi data akun dan kata kunci pada database data autentikasi aktor	R	1
	5. Sistem menmpilkan halaman utama aktor	X	1
	Exception		
	4.a Sistem membaca hasil vaildasi data akun dan kata kunci pada database yang ada pada sistem dan ternyata data akun tersebut tidak valid	W	1

	4.b Sistem kembali ke halaman login	R	1
	4.c Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”	X	1

Tabel 4.54 Re-Measurement Phase UC Ubah Detail Akun Pribadi

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ubah profil” yang terdapat di bawah data akun lama	E	1
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama	X	1
	3. Aktor mengisi data baru pada kolom yang tersedia	E	1
	4. Aktor menekan tombol “Ubah”	E	1
	5. Sistem memvalidasi dan mengubah data akun pribadi terbaru pada database	W	1
	6. Sistem membaca hasil vaildasi data akun pribadi terbaru pada database	R	1
	7. Sistem menampilkan” Halaman Akun Pribadi”	X	1
	Exception		
	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem menampilkan “Halaman Akun Pribadi”	X	1
	6.a Sistem membaca hasil validasi data akun pribadi terbaru pada database tersebut dan ternyata data akun pribadi tersebut tidak valid	R	1
	6.b Sistem menampilkan notifikasi bahwa data akun pribadi salah.	X	1

Setelah proses *re-measurement* ini dilakukan, ternyata seluruh *use case* WikiBudaya yang telah di-*refactoring* tidak ada yang memiliki kecacatan satupun, sehingga dapat ditentukan kembali *use case* apa saja yang terdapat pada WikiBudaya. Berikut merupakan *use case* hasil dari proses *refactoring*:

Tabel 4.55 Use Case hasil dari proses refactoring

Nama Use Case
Daftar Akun
Masuk Akun
Ubah Detail Akun Pribadi
Hapus Akun
Validasi Kontributor
Validasi Reviewer
Ubah Kata Kunci
Tambah Artikel Budaya
Ubah Artikel
Hapus Artikel Budaya
Cari Artikel Budaya
Unggah File
Validasi Artikel
Lihat Histori Artikel
Beri Komentar
Hapus Komentar

Karena terdapat pengurangan *use case* lama setelah dilakukan proses *refactoring*, maka akan dilakukan pula perubahan kode *use case* pada masing – masing *use case* yang ada pada WikiBudaya. Berikut merupakan daftar baru mengenai *use case* dan kode *use case* yang terdapat pada WikiBudaya setelah dilakukan proses identifikasi kecacatan dan proses *refactoring* beserta penggolongan kebutuhan fungsionalnya :

Tabel 4.56 Penjabaran KF-01 baru

Kode Fungsional	Proses Fungsional	Kode
KF-01	Daftar Akun'	UC-01.01
	Masuk Akun	UC-01.02
	Keluar Akun	UC-01.03
	Lihat Detail Akun Pribadi	UC-01.04
	Ubah Detail Akun Pribadi	UC-01.05
	Lihat Data User	UC-01.06
	Hapus Akun	UC-01.07
	Cari User	UC-01.08
	Validasi Kontributor	UC-01.09
	Validasi Reviewer	UC-01.10
	Ubah Kata Kunci	UC-01.11

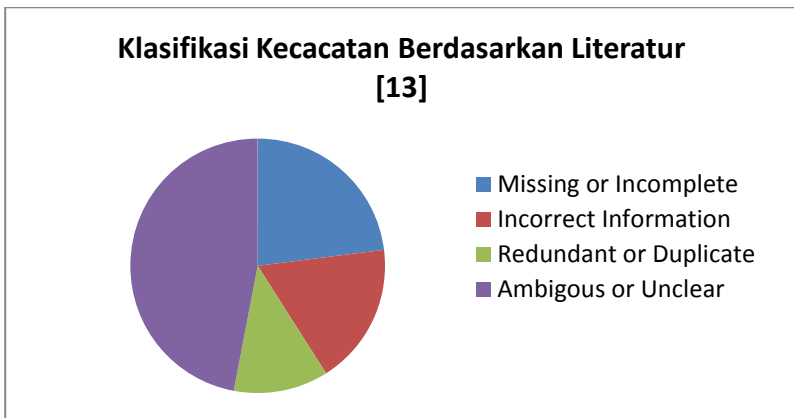
Tabel 4.57 Penjabaran KF-02 baru

Kode Fungsional	Proses Fungsional	Kode
KF-02	Lihat Data Master Artikel Budaya	UC-02.01
	Tambah Artikel Budaya	UC-02.02
	Ubah Artikel Budaya	UC-02.03
	Hapus Artikel Budaya	UC-02.04
	Cari Artikel Budaya	UC-02.05
	Unggah File (lagu,video)	UC-02.06
	Unduh File (lagu,video)	UC-02.07
	Validasi Artikel	UC-02.08
	Lihat Artikel	UC-02.09
	Lihat Status	UC-02.10
	Lihat Histori Artikel	UC-02.11

Tabel 4.58 Penjabaran KF-03 baru

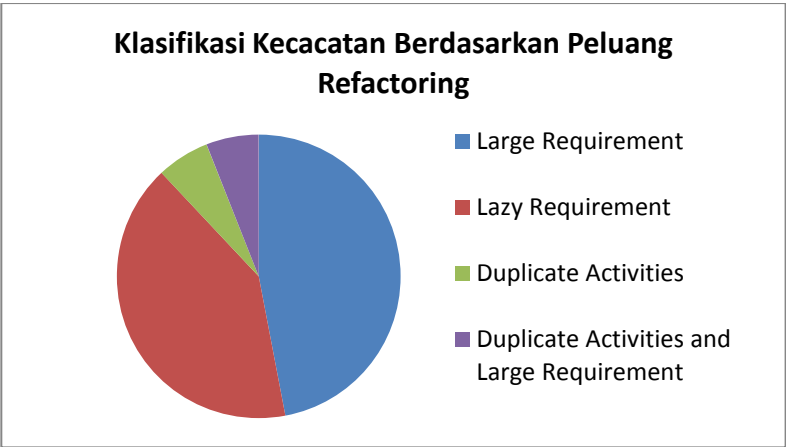
Kode Fungsional	Proses Fungsional	Kode
KF-03	Beri Komentar	UC-03.01
	Lihat Komentar	UC-03.02
	Hapus Komentar	UC-03.03
	Notifikasi Komentar	UC-03.04

Sedangkan berikut merupakan tabel perbandingan mengenai *use case* yang memiliki kecacatan, serta telah diklasifikasi berdasarkan literatur [13] dan peluang *refactoring*.



Gambar 4.8 Gambar Kecacatan Berdasarkan Literatur [13]

Berdasarkan klasifikasi kecacatan menurut literatur [13], diketahui bahwa 23% *use case* yang memiliki kecacatan tergolong pada tipe *missing / incomplete*, sedangkan 18% *use case* tergolong pada tipe *incorrect information* dan 12% *use case* tergolong pada tipe *redundant or duplicate*. Sedangkan sisa 47%-nya tergolong pada tipe *ambiguous or unclear*.



Gambar 4.9 Klasifikasi kecacatan berdasarkan peluang refactoring

Sedangkan berdasarkan peluang *refactoring*, diketahui bahwa terdapat *use case* yang tergolong sebagai *duplicate activities* dan *duplicate activities and large requirement* masing – masing sebesar 6%. Sedangkan Mayoritas *use case* yang memiliki kecacatan tersebut tergolong pada tipe peluang *large requirement* yaitu sebesar 47% dan *lazy requirement* sebesar 41%.

Sedangkan berikut merupakan tabel perbandingan kesesuaian pergerakan data sebelum dan sesudah dilakukan perbaikan :

Tabel 4.59 Tabel Kesesuaian Pergerakan Data Setelah Perbaikan

Kode Use Case	Presentase Kesesuaian Pergerakan Data (sebelum perbaikan)	Jumlah Pergerakan Data (sesudah perbaikan)	Total <i>cfp</i> (sesudah perbaikan)	Presentase Kesesuaian Pergerakan Data (sesudah perbaikan)
UC-01.01	54%	22	22	100%

UC-01.02	75%	8	8	100%
UC-01.03	100%	-	-	100%
UC-01.04	100%	-	-	100%
UC-01.05	44%	11	11	100%
UC-01.06	100%	-	-	100%
UC-01.07	75%	10	10	100%
UC-01.08	100%	-	-	100%
UC-01.09	100%	-	-	100%
UC-01.10	100%	-	-	100%
UC-01.11	100%	-	-	100%
UC-02.01	100%	-	-	100%
UC-02.02	50%	9	9	100%
UC-02.03	50%	9	9	100%
UC-02.04	75%	10	10	100%
UC-02.05	100%	-	-	100%
UC-02.06	50%	11	11	100%
UC-02.07	100%	-	-	100%
UC-02.08	71%	8	8	100%
UC-	100%	-	-	100%

02.09				
UC-02.10	100%	-	-	100%
UC-02.11	100%	-	-	100%
UC-03.01	67%	7	7	100%
UC-03.02	100%	-	-	100%
UC-03.03	57%	10	10	100%
UC-03.04	100%	-	-	100%

Setelah dilakukan perbaikan terhadap *use case* yang memiliki kecacatan dan tidak sesuai terhadap kesesuaian persyaratan pergerakan data, maka didapatkan kumpulan *use case* yang telah sesuai dengan syarat pergerakan data dan tidak meliki kecacatan, sehingga dengan begini *use case* WikiBudaya dapat dikatakan berkualitas sesuai dengan konsep yang telah dijelaskan [6].

BAB 5

PENUTUP

5.1 Kesimpulan

Simpulan yang dapat diambil dari pengerjaan tugas akhir ini adalah sebagai berikut :

1. Berdasarkan identifikasi kondisi awal dari *use case* WikiBudaya, diketahui terdapat 27 *use case* dan didapatkan 17 *use case* memiliki kecacatan.
2. Berdasarkan klasifikasi kecacatan menurut literatur [13], diketahui bahwa 23% *use case* yang memiliki kecacatan tergolong pada tipe *missing / incomplete*, sedangkan 18% *use case* tergolong pada tipe *incorrect information* dan 12% *use case* tergolong pada tipe *redundant or duplicate*. Sedangkan sisa 47%-nya tergolong pada tipe *ambiguous or unclear*.
3. Berdasarkan peluang *refactoring*, diketahui bahwa terdapat *use case* yang tergolong sebagai *duplicate activities* dan *duplicate activities and large requirement* masing – masing sebesar 6%. Sedangkan Mayoritas *use case* yang memiliki kecacatan tersebut tergolong pada tipe peluang *large requirement* yaitu sebesar 47% dan *lazy requirement* sebesar 41%.
4. Setelah didapatkan *use case* WikiBudaya yang memiliki kecacatan dan telah diperbaiki, maka total *use case* WikiBudaya sekarang adalah 26 *use case*. Hal ini dikarenakan adanya *use case* yang memiliki sifat duplikasi, sehingga lebih baik jika digabungkan dengan *use case* yang lain.

5.2 Saran

Saran yang diharapkan dapat dikembangkan di masa mendatang adalah :

1. Dengan adanya berbagai macam metode untuk mengidentifikasi kecacatan pada kebutuhan fungsional suatu perangkat lunak, maka memungkinkan pembaca untuk melakukan identifikasi kecacatan dengan metode yang lainnya selain COSMIC ISO 19761.
2. Dengan adanya berbagai macam metode untuk memperbaiki kecacatan pada kebutuhan fungsional suatu perangkat lunak, maka memungkinkan pembaca untuk melakukan suatu perbaikan pada kebutuhan fungsional dengan metode yang lainnya selain *refactoring*.

BAB 6

DAFTAR PUSTAKA

- [1] Wahono and R. S. romisatriwahono. [Online]. <http://romisatriawahono.net/2006/06/05/teknik-pengukuran-kualitas-perangkat-lunak/>
- [2] Ian Sommerville, "Software Requirement. Software Engineering 7th edition.," 2004.
- [3] R., Piveta, E., Castro, J., Araujo, J., Moreira, A., Guerreiro, P., et al Ramos, "Improving the Quality of Requirements with Refactoring.," Improving the Quality of Requirements with Refactoring., 2009.
- [4] Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them. Firesmith, Donald. S.I. : ETH Zurich, 2007, vol.6.
- [5] S., & Abran, A. Trudel, "Improving quality of functional requirements by measuring their functional size.," Improving quality of functional requirements by measuring their functional size., 2008.
- [6] Universitas Sumatera Utara.
<http://repository.usu.ac.id/bitstream/123456789/30992/3/Capter%2011.pdf>.
- [7] D. Galin, "Software Quality Assurance from Theory. Pearson Education.," Software Quality Assurance from Theory. Pearson Education., 2004.
- [8] C. Melonfire. (2007, Feb.) TechRepublic. [Online]. <http://www.techrepublic.com/article/five-common-errors-in-requirements-analysis-and-how-to-avoid-them/#>.
- [9] K., Proll, B., Reich, S., & Retschitzegger, W. Gerti, "2006," Web Engineering The Discipline of Systematic Development of Web Applications.
- [10] COSMIC, The COSMIC Functional Size Measurement Method. Measurement Manual., 2009.

- [11] Asriningpuri. Prasanti, “Layanan Website Online SAMSAT Surabaya,” Institut Teknologi Sepuluh Nopember., 2014.
- [12] W., & Tso Chen, W Ming Han, Sizing Cloud Applications with ISO/IEC 19761: A case.
- [13] L. M. Isabel, Classification of Defect Types in Requirements Spesification : Literature Review, Proposal and Assesment.
- [14] Baker. cs.fsu.edu. [Online].
www.cs.fsu.edu/~baker%2Fswe1%2Frestricted%2Fnotes%2Fppt%2FUseCaseDiagrams.ppt&ei=hFgrU-6cNc2uiQfFvYCQCw&usg=AFQjCNH-VRrnARK5GTOB8bAgDgWeD68Wfg&sig2=RgPA4SBXifAT1.
- [15] Fitrianto. Nanda, “Analisa Dan Perancangan WikiBudaya dalam Rangka Melestarikan Budaya Bangsa Dan Kearifan Lokal Nusantara,” Institut Teknologi Sepuluh Nopember., 2013.
- [16] Dede. Yunata, “Kakas Bantu Analisis Kerancuan Perngkat Lunak Berbasis Aturan,” Institut Teknologi Sepuluh Nopember.

LAMPIRAN A GENERIC SOFTWARE MODEL

Halaman ini sengaja dikosongkan.

A.1 Generic Software Model

Dibawah ini merupakan tabel *Generic Software Model*, yang merupakan output dari tahap *Mappng Phase*. Tabel ini menjelaskan *sub-process* apa saja yang dimiliki oleh tiap – tiap proses fungsional WikiBudaya.

UC-01.01

Tabel A. 1 UC-01.01 Daftar Akun

Proses Fungsional	Pergerakan Data
Daftar Akun	1. Aktor memilih tombol “Menu”
	2. Sistem menampilkan pilihan “Daftar” dan “Masuk”
	3. Aktor memilih pilihan “Daftar”
	4. Sistem menampilkan <i>form</i> daftar akun (step 1) :
	a) Aktor mengisi <i>form</i> data autentifikasi yang berisikan “Nama Akun” , “Kata Kunci” , “Konfirmasi Kata Kunci” , “Peran”
	b) Apabila nama akun telah digunakan maka sistem akan menjalankan skenario alternatif I No.1
	c) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menjalankan alternatif I No.2

d) Apabila aktor telah mengisikan semua <i>field</i> yang tersedia dan tidak muncul peringatan, aktor melanjutkan skenario 3
5. Aktor menekan tombol :
a) “Lanjut” untuk melanjutkan ke skenario berikutnya
b) “Batal” untuk kembali ke halaman utama
6. Apabila aktor memilih peran sebagai “kontributor”, maka sistem akan menjalankan skenario No.7, dan apabila aktor memilih peran sebagai “reviewer”, maka sistem akan menjalankan skenario alternatif II.
7. Sistem menampilkan halaman data profil (step 2) :
a) Aktor mengisikan <i>formdata</i> profil, yang berisikan “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”pendidikan”, ”pekerjaan” dan “alamat email”
b) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menampilkan alternatif I No.2
8. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
a) “Ok”, maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor.
b) “Kembali”, maka sistem akan menampilkan

halaman daftar sebelumnya (halaman autentifikasi).
Alternatif I
1. Sistem akan menampilkan peringatan (notification), yang berisi bahwa “nama akun telah terdaftar”
2. Sistem akan menampilkan pesan dibawah field yang kosong “Kolom harus diisi.”
Alternatif II
1. Sistem menampilkan halaman data profil (step 2)
a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”bidang studi”, ”pendidikan”, ”pekerjaan”, “alamat email” dan “berkas pendukung”
b) Apabila terdapat field yang kosong, maka sistem akan menampilkan alternatif I No.2
2. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :
a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor
b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi)

UC-01.02**Tabel A. 2 UC-01.02 Masuk Akun**

Proses Fungsional	Pergerakan Data
Masuk Akun	1. Aktor memasukkan “nama akun” dan “kata kunci” pada <i>form</i> masuk
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem
	3. Sistem akan memeriksa apakah <i>form</i> “nama akun” dan “kata kunci” telah sesuai dengan “data autentifikasi aktor”, jika tidak sesuai maka akan dijalankan skenario alternatif I No.1
	Alternatif I
	1. Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”

UC-01.03**Tabel A. 3 UC-01.03 Keluar Akun**

Proses Fungsional	Pergerakan Data
Keluar Akun	1. Aktor memilih menu “Keluar”
	2. Sistem kembali ke halaman utama WikiBudaya

UC-01.04**Tabel A. 4 UC-01.04 Lihat Detail Akun Pribadi**

Proses Fungsional	Pergerakan Data
Lihat Detail Akun Pribadi	1. Aktor memilih menu <i>profil</i>
	2. Sistem menampilkan tampilan data profil dan <i>privilage</i> -nya yang telah tersimpan di database.

UC-01.05**Tabel A. 5 UC-01.05 Ubah Detail Akun Pribadi**

Proses Fungsional	Pergerakan Data
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ <i>ubah profil</i> ” yang terdapat di bawah data akun lama
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama
	3. Aktor mengisi data baru pada kolom yang tersedia
	4. Aktor menekan tombol:
	b) “Ubah”, maka kemudian sistem akan menjalankan skenario sukses utama No.5
	b) “Batal”, maka sistem akan mengembalikan tampilan ke halaman akun pribadi
	5. Sistem melakukan validasi
	c) Benar, maka sistem akan menjalankan skenario utama No.6 dan 7

	d) Salah, maka sistem akan menjalankan skenario alternatif I No.1 dan 2
	6. Sistem akan mengubah data lama dengan data baru ke database
	7. Sistem menampilkan halaman akun

UC-01.06**Tabel A. 6 UC-01.06 Lihat Data User**

Proses Fungsional	Pergerakan Data
Lihat Data User	1. Aktor memilih menu “Daftar akun terdaftar”
	2. Sistem menampilkan tampilan data user secara umum dari masing-masing peran

UC-01.07**Tabel A. 7 UC-01.07 Hapus Akun**

Proses Fungsional	Pergerakan Data
Hapus Akun	1. Aktor memilih data akun yang akan dihapus
	2. Aktor menekan tombol “Hapus” yang terletak disamping nama akun
	3. Sistem akan menampilkan <i>pop up window</i> berupa pertanyaan “Apakah anda yakin untuk menghapus akun ini?”
	4. Aktor menekan tombol :

	a) “Ok”, maka sistem akan menjalankan skenario utama No.4, 5, 6
	b) “Cancel”, maka sistem akan kembali menampilkan halaman daftar user
	5. Sistem akan menghapus akun dari database
	6. Sistem kembali ke halaman daftar akun
	7. Sistem menampilkan notifikasi “akun telah berhasil di hapus

UC-01.08

Tabel A. 8 UC-01.08 Cari User

Proses Fungsional	Pergerakan Data
Cari User	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun

UC-01.09

Tabel A. 9 UC-01.09 Validasi Kontributor

Proses Fungsional	Pergerakan Data
Validasi Kontributor	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian
	2. Sistem menampilkan hasil pencarian berdasarkan

	kata kunci yang dimasukkan, dari semua peran akun
--	---

UC-01.10

Tabel A. 10 UC-01.10 Validasi Reviewer

Proses Fungsional	Pergerakan Data
Validasi Reviewer	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun

UC-01.11

Tabel A. 11 UC-01.11 Ubah Kata Kunci

Proses Fungsional	Pergerakan Data
Ubah Kata Kunci	1. Sistem menampilkan halaman ubah profil
	2. Aktor menekan tombol ubah kata kunci
	3. Sistem menampilkan halaman form ganti kata kunci

UC-02.01**Tabel A. 12 UC-02.01 Lihat Data MAster Artikel Budaya**

Proses Fungsional	Pergerakan Data
Lihat Data Master Artikel Budaya	1. Aktor memilih menu “Daftar Master Artikel”
	2. Sistem menampilkan daftar data master artikel budaya

UC-02.02**Tabel A. 13 UC-02.02 Tambah Artikel Budaya**

Proses Fungsional	Pergerakan Data
Tambah Artikel Budaya	1. Aktor menekan tombol “validate”
	2. Sistem menyimpan artikel budaya yang telah di validasi

UC-02.03**Tabel A. 14 UC-02.03 Ubah Artikel**

Proses Fungsional	Pergerakan Data
Ubah Artikel	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah

	2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah terisi <i>value</i> data lama
	3. Aktor mengisi data baru pada artikel
	4. Aktor menekan tombol :
	a) “simpan”, maka sistem akan menyimpan perubahan dan menjalankan skenario 5
	b) “batal”, maka sistem akan mengembalikan tampilan ke halaman artikel saya
	5. Sistem akan mengirimkan artikel baru ke reviewer

UC-02.04**Tabel A. 15 UC-02.04 Hapus Artikel Budaya**

Proses Fungsional	Pergerakan Data
Hapus Artikel Budaya	1. Aktor memilih artikel yang akan dihapus
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus
	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya ?”
	4. Aktor menekan tombol :
	a) “Ok”, maka sistem akan menjalankan skenario utama nomor 4,5,6
	b) “Batal”, maka sistem akan kembali menampilkan halaman daftar artikel
	5. Sistem akan menghapus artikel dari database

	6. Sistem kembali ke halaman daftar artikel
	7. Sistem menampilkan notifikasi “artikel berhasil dihapus”

UC-02.05

Tabel A. 16 UC-02.05 Cari Artikel Budaya

Proses Fungsional	Pergerakan Data
Cari Artikel Budaya	1. Aktor memasukkan kata / keyword pencarian pada kolom pencarian
	2. Begitu keyword diketikkan, maka sistem akan menampilkan hasil pencarian

UC-02.06

Tabel A. 17 UC-02.06 Unggah File

Proses Fungsional	Pergerakan Data
Unggah File	1. Aktor memilih tombol “Unggah file budaya”.
	2. Sistem menampilkan halaman “pop-up”, yang berisikan “kategori budaya”, “judul file budaya”, dan “tombol browse”
	3. Untuk upload file, ada ketentuan ukuran dan format file yang akan di upload. Untuk ukuran maksimal berukuran 5 Mb, dan format yang

	<p>digunakan haruslah berformat: mp3/mp4, avi. Apabila tidak sesuai maka sistem akan menampilkan alternatif I.no 1, dan apabila format yang digunakan tidak sesuai maka file tidak akan muncul pada file yang di browse</p>
	<p>4. Setelah semua <i>field</i> terisi, dan ukuran file tidak melebihi ukuran yang ditentukan Aktor dapat menekan tombol :</p>
	<p>a) “Ok” ,maka sistem akan mengunggah file budaya ke reviewer dan menjalankan skenario 5</p>
	<p>b) “Batal” untuk membatalkan pengunggahan file budaya</p>
	<p>5. Sistem kembali ke halaman “tuliskan artikel”</p>
	<p>Alternatif I</p>
	<p>1. Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal</p>

UC-02.07

Tabel A. 18 UC-02.07 Unduh File

Proses Fungsional	Pergerakan Data
Unduh File	1. Sistem menampilkan daftar file budaya
	2. Aktor memilih file budaya yang akan diunduh
	3. Aktor menekan tombol “unduh”

UC-02.08**Tabel A. 19 UC-02.08 Review Artikel**

Proses Fungsional	Pergerakan Data
Review Artikel	1. Aktor memilih tombol “Validasi”
	2. Sistem menampilkan halaman artikel yang di review

UC-02.09**Tabel A. 20 UC-02.09 Validasi Artikel**

Proses Fungsional	Pergerakan Data
Validasi Artikel	1. Aktor memilih tombol “ Validasi”
	2. Sistem menampilkan halaman artikel yang di review
	3. Setelah melakukan reviw terhadap artikel budaya, Aktor menekan tombol :
	a) “Setujui”, maka sistem akan menjalankan skenario utama nomor 4,5,6
	b) “Tidak disetujui”, maka sistem akan menampilkan skenario utama nomor 6
	4. Sistem akan menambahkan artikel pada database
	5. Sistem akan menampilkan artikel yang telah divalidasi tersebut
	6. Sistem menampilkan halaman “list artikel”

UC-02.10

Tabel A. 21 UC-02.10 Lihat Artikel

Proses Fungsional	Pergerakan Data
Lihat Artikel	1. Aktor memilih artikel budaya yang akan dilihat (bisa dengan memilih kata kunci atau berdasarkan kategori)
	2. Sistem menampilkan artikel budaya yang telah tersimpan di database
	3. Aktor memilih tombol “Lihat artikel”
	4. Sistem menampilkan artikel budaya

UC-02.11

Tabel A. 22 UC-02.11 Lihat Status Artikel

Proses Fungsional	Pergerakan Data
Lihat Status Artikel	1. Aktor mengetikkan kata kunci artikel
	2. Aktor dapat mengurutkan status artikel berdasarkan yang telah di validasi atau belum di validasi
	3. Sistem menampilkan daftar artikel dan status artikel

UC-02.12**Tabel A. 23 UC-02. 12 Lihat Histori Artikel**

Proses Fungsional	Pergerakan Data
Lihat Histori Artikel	1. Sistem menampilkan history artikel yang pernah di kirimkan oleh kontributor

UC-03.01**Tabel A. 24 UC-03.01 Beri Komentar**

Proses Fungsional	Pergerakan Data
Beri Komentar	1. Aktor mengisikan komentar pada kolom “box komentar”
	2. Sistem menampilkan komentaryang baru dibuat aktor
	3. Sistem menyimpan komentar dari aktor

UC-03.02**Tabel A. 25 UC-03.02 Lihat Komentar**

Proses Fungsional	Pergerakan Data
Lihat Komentar	1. Aktor memilih artikel
	2. Sistem akan menampilkan artikel beserta komentar - komentar yang terdapat di dalamnya

UC-03.03**Tabel A. 26 UC-03.03 Hapus Komentar**

Proses Fungsional	Pergerakan Data
Hapus Komentar	1. Aktor memilih menu komentar
	2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya
	3. Aktor memilih tanda “silang (hapus)” pada komentar yang akan dihapus , aktor akan menekan tombol “ X (silang)” sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini”
	4. Aktor menekan tombol :
	a) “ OK “ sistem akan menghapus komentar dan menjalankan skenario 5.6
	b) “ Batal” Sistem menjalankan skenario 6
	5. Sistem memperbaharui database komentar
	6. Sistem menampilkan halaman komentar

UC-03.04**Tabel A. 27 UC-03.04 Notifikasi Komentar**

Proses Fungsional	Pergerakan Data
Notifikasi Komentar	1. Sistem menampilkan notifikasi komentar baru
	2. Admin menekan notifikasi komentar
	3. Sistem menampilkan komentar beserta artikel

LAMPIRAN B
MEASUREMENT PHASE TABLE

Halaman ini sengaja dikosongkan.

UC-01.01

Tabel B. 1 Analisa Cacat UC-01.01

Proses Fungsional	Pergerakan Data	Tipe	CFP
Daftar Akun	1. Aktor memilih tombol “Menu”	E	1
	2. Sistem menampilkan pilihan “ Daftar” dan “Masuk”	X	1
	3. Aktor memilih pilihan “Daftar”	E	1
	4. Sistem menampilkan <i>form</i> daftar akun (step 1) :	X	1
	a) Aktor mengisi <i>form</i> data autentifikasi yang berisikan “Nama Akun” , “Kata Kunci” , “Konfirmasi Kata Kunci” , “Peran”	E	1
	b) Apabila nama akun telah digunakan maka sistem akan menjalankan skenario alternatif I No.1		0
	c) Apabila terdapat <i>field</i> yang kosong, maka sistem akan menjalankan alternatif I No.2		0
	d) Apabila aktor telah mengisi semua <i>field</i> yang tersedia dan tidak muncul peringatan, aktor melanjutkan skenario 3		0
	5. Aktor menekan tombol :		
	a) “Lanjut” untuk melanjutkan ke skenario berikutnya	E	1
	b) “Batal” untuk kembali ke halaman utama	E	1
	6. Apabila aktor memilih peran sebagai “kontributor”, maka sistem akan menjalankan skenario No.7, dan apabila aktor memilih peran sebagai “reviewer”,		0

maka sistem akan menjalankan skenario alternatif II.		
7. Sistem menampilkan halaman data profil (step 2) :	X	1
a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”pendidikan”, ”pekerjaan” dan “alamat email”	E	1
b) Apabila terdapat <i>field</i> yang kosong,maka sistem akan menampilkan alternatif I No.2		0
8. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :		
a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor.		0
b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi).		0
Alternatif I		
1. Sistem akan menampilkan peringatan (notification), yang berisi bahwa “nama akun telah terdaftar”	X	1
2. Sistem akan menampilkan pesan dibawah field yang kosong “Kolom harus diisi.”	X	1
Alternatif II		
1. Sistem menampilkan halaman data profil (step 2)	X	1
a) Aktor mengisikan <i>form</i> data profil, yang berisikan “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”bidang studi”, ”pendidikan”, ”pekerjaan”, “alamat	E	1

	email” dan “berkas pendukung”		
	b) Apabila terdapat field yang kosong, maka sistem akan menampilkan alternatif I No.2		0
	2. Apabila semua <i>field</i> telah terisi, aktor menekan tombol :		
	a) “Ok” , maka sistem akan menampilkan halaman awal dari peran yang dipilih oleh aktor		0
	b) “Kembali” , maka sistem akan menampilkan halaman daftar sebelumnya (halaman autentifikasi)		0

UC-01.02**Tabel B. 2 Analisa Cacat UC-01.02**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Masuk Akun	1. Aktor memasukkan “nama akun” dan “kata kunci” pada <i>form</i> masuk	E	1
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam system	E	1
	3. Sistem akan memeriksa apakah <i>form</i> “nama akun” dan “kata kunci” telah sesuai dengan “data autentifikasi aktor”, jika tidak sesuai maka akan dijalankan skenario alternatif I No.1		0
	Alternatif I		
	1. Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”	X	1

UC-01.03**Tabel B. 3 Analisa Cacat UC-01.03**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Keluar Akun	1. Aktor memilih menu “Keluar”	E	1
	2. Sistem kembali ke halaman utama WikiBudaya	X	1

UC-01.04**Tabel B. 4 Analisa Cacat UC-01.04**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Detail Akun Pribadi	1. Aktor memilih menu <i>profil</i>	E	1
	2. Sistem menampilkan tampilan data profil dan <i>privilage</i> -nya yang telah tersimpan di database.	X	1

UC-01.05**Tabel B. 5 Analisa Cacat UC-01.05**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Detail Akun Pribadi	1. Aktor menekan tombol “ <i>ubah profil</i> ” yang terdapat di bawah data akun lama	E	1
	2. Sistem akan menampilkan halaman ubah user dengan kolom yang sudah terisi value data lama	X	1
	3. Aktor mengisi data baru pada kolom yang tersedia	E	1
	4. Aktor menekan tombol :		

	a) “Ubah”, maka kemudian sistem akan menjalankan skenario sukses utama No.5		0
	b) “Batal”, maka sistem akan mengembalikan tampilan ke halaman akun pribadi		0
	5. Sistem melakukan validasi		
	a) Benar, maka sistem akan menjalankan skenario utama No.6 dan 7		0
	b) Salah, maka sistem akan menjalankan skenario alternatif I No.1 dan 2		0
	6. Sistem akan mengubah data lama dengan data baru ke database	W	0
	7. Sistem menampilkan halaman akun	X	1

UC-01.06

Tabel B. 6 Analisa Cacat UC-01.06

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Data User	1. Aktor memilih menu “Daftar akun terdaftar”	E	1
	2. Sistem menampilkan tampilan data user secara umum dari masing-masing peran	X	1

UC-01.07

Tabel B. 7 Analisa Cacat UC-01.07

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Akun	1. Aktor memilih data akun yang akan dihapus	E	1
	2. Aktor menekan tombol “Hapus” yang	E	1

	terletak disamping nama akun		
	3. Sistem akan menampilkan <i>pop up window</i> berupa pertanyaan “Apakah anda yakin untuk menghapus akun ini?”	X	1 1
	4. Aktor menekan tombol :		
	a) “Ok”, maka sistem akan menjalankan skenario utama No.4, 5, 6		0
	b) “Cancel”, maka sistem akan kembali menampilkan halaman daftar user		0
	5. Sistem akan menghapus akun dari database	W	1
	6. Sistem kembali ke halaman daftar akun	X	1
	7. Sistem menampilkan notifikasi “akun telah berhasil di hapus	X	1

UC-01.08

Tabel B. 8 Analisa Cacat UC-01.08

Proses Fungsional	Pergerakan Data	Tipe	CFP
Cari User	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian	E	1
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun	X	1

UC-01.09

Tabel B. 9 Analisa Cacat UC-01.09

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi	1. Aktor memasukkan kata kunci berupa	E	1

Kontributor	nama dari pengguna pada kolom pencarian		
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun	X	1

UC-01.10**Tabel B. 10 Analisa Cacat UC-01.10**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi Reviewer	1. Aktor memasukkan kata kunci berupa nama dari pengguna pada kolom pencarian	E	1
	2. Sistem menampilkan hasil pencarian berdasarkan kata kunci yang dimasukkan, dari semua peran akun	X	1

UC-01.11**Tabel B. 11 Analisa Cacat UC-01.11**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Kata Kunci	1. Sistem menampilkan halaman ubah profil	X	1
	2. Aktor menekan tombol ubah kata kunci	E	1
	3. Sistem menampilkan halaman form ganti kata kunci	X	1

UC-02.01**Tabel B. 12 Analisa Cacat UC-02.01**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Data Master Artikel Budaya	1. Aktor memilih menu “Daftar Master Artikel”	E	1
	2. Sistem menampilkan daftar data master artikel budaya	X	1

UC-02.02**Tabel B. 13 Analisa Cacat UC-02.02**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Tambah Artikel Budaya	1. Aktor menekan tombol “validate”	E	1
	2. Sistem menyimpan artikel budaya yang telah di validasi		0

UC-02.03**Tabel B. 14 Analisa Cacat UC-02.03**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Artikel	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah	E	1
	2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah terisi <i>value</i> data lama	X	1
	3. Aktor mengisi data baru pada artikel	E	1
	4. Aktor menekan tombol :		

	a) “simpan”, maka sistem akan menyimpan perubahan dan menjalankan skenario 5		0
	b) “batal”, maka sistem akan mengembalikan tampilan ke halaman artikel saya		0
	5. Sistem akan mengirimkan artikel baru ke reviewer		0

UC-02.04**Tabel B. 15 Analisa Cacat UC-02.04**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Artikel Budaya	1. Aktor memilih artikel yang akan dihapus	E	1
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus	E	1
	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya?”	X	1
	4. Aktor menekan tombol :		
	a) “Ok”, maka sistem akan menjalankan skenario utama nomor 4,5,6		0
	b) “Batal”, maka sistem akan kembali menampilkan halaman daftar artikel		0
	5. Sistem akan menghapus artikel dari database	W	1
	6. Sistem kembali ke halaman daftar artikel	R	1
	7. Sistem menampilkan notifikasi “artikel berhasil dihapus”	X	1

UC-02.05

Tabel B. 16 Analisa Cacat UC-02.05

Proses Fungsional	Pergerakan Data	Tipe	CFP
Cari Artikel Budaya	1. Aktor memasukkan kata / keyword pencarian pada kolom pencarian	E	1
	2. Begitu keyword diketikkan, maka sistem akan menampilkan hasil pencarian	X	1

UC-02.06

Tabel B. 17 Analisa Cacat UC-02.06

Proses Fungsional	Pergerakan Data	Tipe	CFP
Unggah File	1. Aktor memilih tombol “Unggah file budaya”.	E	1
	2. Sistem menampilkan halaman “pop-up” , yang berisikan “kategori budaya”, “judul file budaya”, dan “tombol browse”	X	1
	3. Untuk upload file, ada ketentuan ukuran dan format file yang akan di upload. Untuk ukuran maksimal berukuran 5 Mb, dan format yang digunakan haruslah berformat: mp3/mp4, avi. Apabila tidak sesuai maka sistem akan menampilkan alternatif I.no 1, dan apabila format yang digunakan tidak sesuai maka file tidak akan muncul pada file yang di browse		0
	4. Setelah semua <i>field</i> terisi, dan ukuran file tidak melebihi ukuran yang ditentukan Aktor dapat menekan tombol :		0
	a) “Ok” ,maka sistem akan		0

	mengunggah file budaya ke reviewer dan menjalankan skenario 5		
	b) “Batal” untuk membatalkan pengunggahan file budaya		0
	5. Sistem kembali ke halaman “tuliskan artikel”	X	1
	Alternatif I		
	1. Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal	X	1

UC-02.07

Tabel B. 18 Analisa Cacat UC-02.07

Proses Fungsional	Pergerakan Data	Tipe	CFP
Unduh File	1. Sistem menampilkan daftar file budaya	X	1
	2. Aktor memilih file budaya yang akan diunduh	E	1
	3. Aktor menekan tombol “unduh”	E	1

UC-02.08

Tabel B. 19 Analisa Cacat UC-02.08

Proses Fungsional	Pergerakan Data	Tipe	CFP
Review Artikel	1. Aktor memilih tombol “Validasi”	E	1
	2. Sistem menampilkan halaman artikel yang di review	X	1

UC-02.09**Tabel B. 20 Analisa Cacat UC-02.09**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi Artikel	1. Aktor memilih tombol “ Validasi”	E	1
	2. Sistem menampilkan halaman artikel yang di review	X	1
	3. Setelah melakukan review terhadap artikel budaya, Aktor menekan tombol :		
	a) “Setujui”, maka sistem akan menjalankan skenario utama nomor 4,5,6		0
	b) “Tidak Disetujui”, maka sistem akan menampilkan skenario utama nomor 6		0
	4. Sistem akan menambahkan artikel pada database	W	1
	5. Sistem akan menampilkan artikel yang telah divalidasi tersebut	R	1
	6. Sistem menampilkan halaman “list artikel”	X	1

UC-02.10**Tabel B. 21 Analisa Cacat UC-02.10**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Artikel	1. Aktor memilih artikel budaya yang akan dilihat (bisa dengan memilih kata kunci atau berdasarkan kategori)	E	1
	2. Sistem menampilkan artikel budaya yang telah tersimpan di database	X	1
	3. Aktor memilih tombol “Lihat artikel”	E	1
	4. Sistem menampilkan artikel budaya	X	1

UC-02.11**Tabel B. 22 Analisa Cacat UC-02.11**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Status Artikel	1. Aktor mengetikkan kata kunci artikel	E	1
	2. Aktor dapat mengurutkan status artikel berdasarkan yang telah di validasi atau belum di validasi	E	1
	3. Sistem menampilkan daftar artikel dan status artikel	X	1

UC-02.12**Tabel B. 23 Analisa Cacat UC-02.12**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Histori Artikel	1. Sistem menampilkan <i>history</i> artikel yang pernah di kirimkan oleh kontributor	X	1

UC-03.01**Tabel B. 24 Analisa Cacat UC-03.01**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Beri Komentar	1. Aktor mengisikan komentar pada kolom “ box komentar”	E	1
	2. Sistem menampilkan komentar yang baru dibuat aktor	X	1
	3. Sistem menyimpan komentar dari aktor	W	0

UC-03.02**Tabel B. 25 Analisa Cacat UC-03.02**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Komentar	1. Aktor memilih artikel	E	1
	2. Sistem akan menampilkan artikel beserta komentar - komentar yang terdapat di dalamnya	X	1

UC-03.03**Tabel B. 26 Analisa Cacat UC-03.03**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Komentar	1. Aktor memilih menu komentar	E	1
	2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya	X	1
	3. Aktor memilih tanda “silang (hapus)” pada komentar yang akan dihapus, aktor akan menekan tombol “ X (silang)” sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini”		0
	4. Aktor menekan tombol :		
	a) “ OK “ sistem akan menghapus komentar dan menjalankan skenario 5.6		0
	b) “ Batal” Sistem menjalankan skenario 6		0
	5. Sistem memperbaharui database komentar	W	1
	6. Sistem menampilkan halaman komentar	X	1

UC-03.04**Tabel B. 27 Analisa Cacat UC-03.04**

Proses Fungsional	Pergerakan Data	Tipe	CFP
Notifikasi Komentar	1. Sistem menampilkan notifikasi komentar baru	X	1
	2. Admin menekan notifikasi komentar	E	1
	3. Sistem menampilkan komentar beserta artikel	X	1

Halaman ini sengaja dikosongkan.

LAMPIRAN C

USE CASE HASIL REFACTORING

Halaman ini sengaja dikosongkan.

C.1 Use Case Hasil Refactoring

Dibawah ini merupakan hasil dari proses perbaikan menggunakan refactoring. Ada beberapa use case baru hasil dari perbaikan menggunakan refactoring, diantaranya:

Tabel C. 1 UC-01.01 Daftar Akun

+	UC-01.01 Daftar Akun	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor memiliki akun untuk mendapatkan hak akses ke WikiBudaya	
	Direct Aktor :	<i>Kontributor, Reviewer</i>
Relasi antar use-case	-	
Trigger :	Aktor ingin menjadi kontributor / <i>reviewer</i> pada WikiBudaya	
Prioritas :	Penting (<i>essential</i>)	
Frekuensi Penggunaan :	Sekali saja setiap user (<i>once</i>)	
Pre Condition :	Aktor telah masuk halaman awal WikiBudaya.	
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Menu” 2. Sistem menampilkan pilihan “Daftar” dan “Masuk” 3. Aktor memilih pilihan “Daftar” 4. Sistem menampilkan <i>form</i> “Daftar Akun” 	

5. Aktor mengisikan *form* data profil, yang berisikan “nama akun” , ”kata kunci” , ”konfirmasi kata kunci” dan “peran”
6. Aktor memilih peran sebagai kontributor / *reviewer*
7. Aktor menekan tombol “Lanjut”
8. Sistem memvalidasi dan memproses data akun pada database kontributor / *reviewer*
9. Sistem membaca hasil vaildasi data akun pada database kontributor / *reviewer*
10. Sistem menampilkan halaman *form* “Data Profil”
11. Aktor mengisi semua *field* pada data profil tentang “nama lengkap” , “jenis kelamin” , ”tanggal lahir” , ”pendidikan” , ”pekerjaan” dan “alamat email”
12. Aktor menekan tombol “Selesai”
13. Sistem memvalidasi dan memproses data profil pada database kontributor / *reviewer*
14. Sistem membaca hasil vaildasi data profil pada database kontributor / *reviewer*
15. Sistem menampilkan halaman utama kontributor / *reviewer*

<p>Alternative Flows</p>	<p>1. Jika memilih sebagai <i>reviewer</i>, maka aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap”, “jenis kelamin”, “tanggal lahir”, “bidang studi”, “pendidikan”, “pekerjaan”, “alamat email” dan “berkas penunjang”</p>
<p>Exception :</p>	<p>9.a Sistem membaca hasil vaildasi data akun pada database kontributor / <i>reviewer</i> dan ternyata ada data akun yang tidak valid</p> <p>9.b Sistem kembali menampilkan <i>form</i> “Daftar Akun”</p> <p>9.c Sistem akan menampilkan peringatan, yang berisi bahwa “Nama Akun Telah Terdaftar” atau “Semua Kolom Harus Diisi”</p> <p>14.a Sistem membaca hasil vaildasi data profil pada database kontributor / <i>reviewer</i> dan ternyata ada data profil yang tidak valid</p> <p>14.b Sistem kembali menampilkan <i>form</i> “Data Profil”</p> <p>14.c Sistem akan menampilkan peringatan, yang berisi bahwa “Semua Kolom Harus Diisi”</p>

Tabel C. 2 UC-01.02 Masuk Akun

+	UC-01.02 Masuk Akun
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat masuk ke dalam aplikasi, setelah sebelumnya melakukan registrasi (daftar akun)
Direct Aktor :	Admin sistem, admin reviewer, reviewer, kontributor
Relasi Antar Use-case :	-
Trigger :	Aktor ingin masuk ke dalam akun WikiBudaya-nya
Prioritas :	Penting (<i>essential</i>)
Frekuensi Penggunaan :	Sekali saja setiap user (<i>once</i>)
Pre Condition :	Aktor telah masuk halaman login, sesuai dengan peran aktor.
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor memasukkan “Nama Akun” dan “Kata Kunci” pada <i>form</i> masuk 2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem 3. Sistem mengecek dan memvalidasi data akun dan kata kunci pada database yang ada pada sistem 4. Sistem membaca hasil validasi data akun dan kata kunci pada database yang ada pada sistem 5. Sistem menampilkan halaman utama aktor
Exception :	4.a Sistem membaca hasil validasi data

	<p>akun dan kata kunci pada database yang ada pada sistem dan ternyata data akun tersebut tidak valid</p> <p>4.b Sistem kembali ke halaman login</p> <p>4.c Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”</p>
--	--

Tabel C. 3 UC-01.05 Ubah Detail Akun Pribadi

+	UC-01.05 Ubah Detail Akun Pribadi	
	Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat mengubah data akun yang dimilikinya sendiri
	Direct Aktor :	Admin sistem, admin reviewer, reviewer, kontributor
	Relasi antar use-case :	Extend (UC-01.04)
	Trigger :	Aktor ingin mengubah detail dari akunnya
	Prioritas :	Penting (<i>essential</i>)
	Frekuensi Penggunaan :	Jarang (<i>rarely</i>)
	Pre Condition :	Aktor telah masuk di halaman profil pengguna
	Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor menekan tombol “ubah profil” yang terdapat di bawah data akun lama 2. Sistem akan menampilkan halaman

Exception :	ubah user dengan kolom yang sudah terisi value data lama 3. Aktor mengisi data baru pada kolom yang tersedia 4. Aktor menekan tombol “Ubah” 5. Sistem memvalidasi dan mengubah data akun pribadi terbaru pada database yang ada pada sistem 6. Sistem membaca hasil validasi data akun pribadi terbaru pada database yang ada pada sistem 7. Sistem menampilkan” Halaman Akun Pribadi”
	4.a Jika aktor menekan “Batal” 4.b Sistem menampilkan “Halaman Akun Pribadi” 6.a Sistem membaca hasil validasi data akun pribadi terbaru pada database tersebut dan ternyata data akun pribadi tersebut tidak valid 6.b Sistem menampilkan notifikasi bahwa data akun pribadi salah.

Tabel C. 4 UC-01.07 Hapus Akun

+	UC-01.07 Hapus Akun
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat menghapus akun user yang dikarenakan user tidak aktif selama 3 tahun.

Direct Aktor :	Admin sistem
Relasi antar use-case :	Extend (UC-01.06)
Trigger :	Aktor ingin menghapus akun dari user yang tidak aktif
Prioritas :	Diharapkan
Frekuensi Penggunaan :	Jarang (<i>rarely</i>)
Pre Condition :	Aktor telah masuk di database akun
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor memilih data akun yang akan dihapus 2. Aktor menekan tombol “Hapus” yang terletak disamping nama akun 3. Sistem akan menampilkan <i>pop up window</i> berupa pertanyaan “Apakah anda yakin untuk menghapus akun ini?” 4. Aktor menekan tombol “Ok” 5. Sistem akan menghapus akun dari database yang ada pada sistem 6. Sistem akan membaca hasil pembaharuan database terebut 7. Sistem kembali ke halaman “daftar akun” 8. Sistem menampilkan notifikasi “akun telah berhasil di hapus”
Exception :	<ol style="list-style-type: none"> 4.a Jika aktor menekan “Batal” 4.b Sistem kembali ke halaman “daftar akun”

Tabel C. 5 UC-01.09 Validasi Kontributor

+	UC-01.09 Validasi Kontributor	
Ringkasan :		<i>Use-case</i> ini dimaksudkan agar aktor dapat memvalidasi akun dari user yang mendaftar sebagai kontributor
Direct Aktor :		Admin sistem
Relasi antar use-case :		-
Trigger :		Aktor ingin memvalidasi akun dari user yang mendaftar sebagai kontributor
Prioritas :		Penting (<i>essential</i>)
Frekuensi Penggunaan :		Jarang (<i>rarely</i>)
Pre Condition :		Aktor telah berada di halaman login situs admin WikiBudaya
Skenario Sukses Utama :		<ol style="list-style-type: none">1. Aktor login ke dalam situs WikiBudaya2. Sistem menampilkan halaman utama admin sistem dan menampilkan pesan ada pengguna baru3. Aktor membuka detail pengguna4. Sistem menampilkan detail pengguna5. Aktor menekan tombol “Validasi”6. Sistem kembali ke halaman utama admin sistem
Skenario Alternatif :		-

Tabel C. 6 UC-01.10 Validasi Reviewer

+	UC-01.10 Validasi Reviewer	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat memvalidasi akun dari user yang mendaftar sebagai <i>reviewer</i>	
Direct Aktor :	Admin <i>Reviewer</i>	
Relasi antar use-case :	-	
Trigger :	Aktor ingin memvalidasi akun dari user yang mendaftar sebagai <i>reviewer</i>	
Prioritas :	Penting (<i>essential</i>)	
Frekuensi Penggunaan :	Jarang (<i>rarely</i>)	
Pre Condition :	Aktor telah berada di halaman login situs admin WikiBudaya	
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor login ke dalam situs WikiBudaya 2. Sistem menampilkan halaman utama admin <i>reviewer</i> dan menampilkan pesan ada <i>reviewer</i> baru 3. Aktor membuka detail <i>reviewer</i> 4. Sistem menampilkan detail pengguna 5. Aktor menekan tombol “Validasi” 6. Sistem kembali ke halaman utama admin <i>reviewer</i> 	
Skenario Alternatif :	-	

Tabel C. 7 UC-01.11 Ubah kata kunci

+	UC-01.11 Ubah kata kunci	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat mengganti kata kunci dari akunnya	
Direct Aktor :	Admin sistem, admin reviewer, kontributor, reviewer	
Relasi antar use-case :	Extend (UC-01.04)	
Trigger :	Aktor ingin mengubah kata kunci akun WikiBudaya	
Prioritas :	Penting (<i>essential</i>)	
Frekuensi Penggunaan :	Jarang (<i>rarely</i>)	
Pre Condition :	Aktor telah berada di halaman login situs WikiBudaya	
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor login ke dalam situs WikiBudaya 2. Sistem menampilkan halaman dari profil pengguna 3. Aktor menekan tombol ubah kata kunci 4. Sistem menampilkan halaman untuk ubah kata kunci 5. Aktor mengisi field untuk mengubah kata kunci 6. Aktor menekan tombol “Simpan” 7. Sistem memvalidasi dan mengubah data kata kunci terbaru pada database yang ada pada sistem 8. Sistem membaca hasil validasi data kata kunci terbaru pada database tersebut 9. Sistem kembali menampilkan halaman 	

Exception :	dari profil pengguna
	7.a Sistem membaca hasil validasi data kata kunci terbaru pada database tersebut dan ternyata data kata kunci tersebut tidak valid 7.b Sistem menampilkan notifikasi bahwa kata kunci salah.

Tabel C. 8 UC-02.02 Tambah Artikel Budaya

+	UC-02.02 Tambah Artikel Budaya	
Ringkasan :	<i>Use-case</i> ini dimaksudkan penambahan artikel budaya pada WikiBudaya. Artikel dapat bertambah, apabila artikel telah divalidasi oleh pihak reviewer.	
Direct Aktor :	Kontributor	
Relasi antar use-case :	-	
Trigger :	Aktor ingin menambah artikel pada WikiBudaya	
Prioritas :	Penting	
Frekuensi Penggunaan :	Selalu (<i>always</i>)	
Pre Condition :	Aktor telah masuk di halaman home dari aktor	
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor menekan <i>tab menu</i> “Tulis Artikel” 2. Sistem akan menampilkan halaman untuk membuat artikel baru 3. Aktor menuliskan artikel baru pada <i>field</i> yang tersedia 	

Exception :	4. Aktor menekan tombol “Simpan” 5. Sistem menyimpan artikel baru dari aktor pada database artikel 6. Sistem membaca artikel baru dari aktor yang telah disimpan pada database artikel 7. Sistem menampilkan halaman “Daftar Master Artikel”
	4.a Jika aktor menekan “Batal” 4.b Sistem menampilkan halaman “Daftar Master artikel”

Tabel C. 9 UC-02.03 Ubah Artikel

+	UC-02.03 Ubah Artikel	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat mengubah artikel	
Direct Aktor :	Kontributor	
Relasi antar use-case :	-	
Trigger :	Aktor ingin mengubah artikel yang pernah dibuat	
Prioritas :	Diharapkan	
Frekuensi Penggunaan :	Jarang (<i>rarely</i>)	
Pre Condition :	Aktor telah masuk di halaman artikel saya	
Skenario Sukses Utama :	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah 2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah	

Exception :	terisi <i>value</i> data lama
	3. Aktor mengisi data baru pada artikel 4. Aktor menekan tombol “simpan” 5. Sistem akan mengirimkan artikel baru ke database artikel 6. Sistem membaca data artikel yang telah diperbaharui pada database artikel 7. Sistem kembali ke halaman “artikel saya”
Exception :	4.a Jika aktor menekan “Batal”
	4.b Sistem kembali ke halaman “artikel saya”Artikel

Tabel C. 10 UC-02.09 Hapus Artikel

+	UC-02.04 Hapus Artikel Budaya	
Ringkasan :		<i>Use-case</i> ini dimaksudkan agar aktor dapat menghapus data artikel budaya yang diminta oleh reviewer dan kontributor.
Direct Aktor :		Admin reviewer
Relasi antar use-case :		Extend (UC-02.01)
Trigger :		Aktor ingin mengahapus artikel budaya pada WikiBudaya
Prioritas :		Penting
Frekuensi Penggunaan :		Jarang (<i>rarely</i>)
Pre Condition :		Aktor telah masuk di halaman “ Daftar Master Artikel”
Skenario Sukses		1. Aktor memilih artikel yang akan

Utama :	dihapus
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus
Exception :	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya?”
	4. Aktor menekan “Ok”
Exception :	5. Sistem akan menghapus artikel dari database artikel
	6. Sistem membaca database artikel yang telah diperbaharui
Exception :	7. Sistem kembali ke halaman “daftar artikel”
	8. Sistem menampilkan notifikasi “artikel berhasil dihapus”
Exception :	4.a Jika aktor menekan “Batal”
	4.b Sistem kembali ke halaman “daftar artikel”

Tabel C. 11 UC-02.04 Cari Artikel

+	UC-02.05 Cari Artikel Budaya	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat mencari data artikel budaya	
	Direct Aktor : Admin sistem, admin reviewer, kontributor, reviewer, user-umum	
Relasi antar use-case		Include (UC-02.09)
Trigger :		Aktor ingin mencari suatu data artikel budaya pada WikiBudaya

Prioritas :	Penting
Frekuensi Penggunaan :	Selalu (<i>Always</i>)
Pre Condition :	Aktor telah masuk di halaman artikel saya / home dari aktor
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor memasukkan kata / <i>keyword</i> pencarian pada kolom pencarian 2. Aktor menekan tombol “Cari Budaya” 3. Maka sistem akan menampilkan hasil pencarian
Skenario Alternatif :	-

Tabel C. 12 UC-02.06 Unggah File

+	UC-02.06 Unggah File	
Ringkasan :	<i>Use-case</i> ini dimaksudkan untuk mengunggah file (lagu dan video) yang telah dibuat oleh kontributor untuk di kirimkan ke pihak reviewer.	
Direct Aktor :	Kontributor	
Relasi antar use-case :	-	
Trigger :	Aktor ingin mengunggah suatu file pada WikiBudaya	
Prioritas :	Penting	
Frekuensi Penggunaan :	Sering	

Pre Condition :	Aktor telah masuk di halaman Tulis Artikel
Skenario Sukses Utama :	<ol style="list-style-type: none">1. Aktor memilih tombol “ Unggah file budaya”2. Sistem menampilkan halaman “Ulpoad”3. Aktor meng-<i>upload file</i> sesuai dengan ketentuan ukuran dan format <i>file</i>4. Aktor menekan “upload”5. Sistem mengunggah file tersebut ke dalam database artikel6. Sistem membaca file yang telah diunggah pada database artikel7. Sistem kembali ke halaman “tulis artikel”
Exception :	<ol style="list-style-type: none">3.a Aktor meng-<i>upload file</i> tidak sesuai dengan ketentuan ukuran dan format <i>file</i>3.b File tidak akan muncul pada file yang di browse Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal5.a Jika aktor menekan “Batal”5.b Sistem kembali ke halaman “tulis artikel”

Tabel C. 13 UC-02.08 Validasi artikel

+	UC-02.08 Validasi artikel	
Ringkasan :		<i>Use-case</i> ini dimaksudkan agar aktor dapat meng-validasi artikel yang dianggap layak oleh pihak reviewer, apabila artikel telah divalidasi maka artikel secara otomatis akan
Direct Aktor :		Reviewer

Relasi antar use-case :	-
Trigger :	Aktor ingin memvalidasi artikel yang layak
Prioritas :	Penting
Frekuensi Penggunaan :	Sering
Pre Condition :	Aktor telah masuk di halaman list artikel
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor memilih tombol “Validasi” 2. Sistem menampilkan halaman artikel yang di <i>review</i> 3. Aktor menekan tombol “Setujui” 4. Sistem akan menambahkan artikel pada database 5. Sistem akan menampilkan artikel yang telah divalidasi tersebut 6. Sistem menampilkan halaman “list artikel”
Exception :	3.a Jika aktor menekan “Tidak Disetujui” 3.b Sistem menampilkan halaman “list artikel”

Tabel C. 14 UC-02.11 Lihat History Artikel

+	UC-02.11 Lihat History Artikel
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat melihat artikel-artikel yang sebelumnya pernah dikirimkan oleh pihak kontributor dan yang telah divalidasi oleh reviewer.
Relasi antar use-case :	-

Direct Aktor :	Reviewer. kontributor
Trigger :	Aktor ingin melihat rekapan artikel yang telah divalidasi
Prioritas :	Penting (<i>essential</i>)
Frekuensi Penggunaan :	Jarang (<i>rarely</i>)
Pre Condition :	Aktor telah masuk di halaman utama kontributor / <i>reviewer</i>
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor menekan <i>tab menu</i> “Akitiftas Histori” pada halaman utama kontributor 2. Sistem menampilkan <i>history</i> artikel yang pernah di kirimkan oleh kontributor
Skenario Alternatif :	-

Tabel C. 15 UC-03.01Beri Komentar

UC-03.01 Beri Komentar	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat memberikan atau mengisikan kometar tentang budaya
Direct Aktor :	Admin reviewer, reviewer, kontributor.
Relasi antar use-case :	Extend (UC-03.02)
Prioritas :	Penting

Frekuensi Penggunaan :	Sering (<i>always</i>)
Pre Condition :	Aktor telah masuk di halaman suatu artikel
Skenario Sukses Utama :	<ol style="list-style-type: none"> 1. Aktor menekan tab menu “komentar” 2. Sistem menampilkan form komentar 3. Aktor mengisi komentar pada kolom “box komentar”. 4. Aktor menekan tombol “Kirim” 5. Sistem menyimpan komentar dari aktor pada database komentar 6. Sistem membaca komentar dari aktor yang telah disimpan pada database komentar 7. Sistem menampilkan komentar yang baru dibuat aktor.
Skenario Alternatif :	-

Tabel C. 16 UC-03.03 Hapus Komentar

UC-03.03 Hapus Komentar	
Ringkasan :	<i>Use-case</i> ini dimaksudkan agar aktor dapat menghapus komentar yang di rasa tidak layak pada artikel budaya.
Direct Aktor :	Admin reviewer
Relasi antar use-case :	Extend (UC-03.02)
Trigger :	Aktor ingin menghapus komentar dari suatu artikel budaya

Prioritas : Frekuensi Penggunaan : Pre Condition : Skenario Sukses Utama : Exception :	Penting
	Jarang
	Aktor telah masuk di halaman artikel
	<ol style="list-style-type: none"> 1. Aktor memilih menu komentar 2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya 3. Aktor menekan tanda “silang (hapus)” pada komentar yang akan dihapus 4. Sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini” 5. Aktor menekan “OK” 6. Sistem memperbaharui database komentar 7. Sistem membaca data komentar yang telah diperbaharui pada database komentar 8. Sistem menampilkan halaman komentar
	5.a Jika aktor menekan “Batal”
	5.b Sistem kembali menampilkan halaman komentar

LAMPIRAN D
RE-MEASUREMENT PHASE TABLE

Halaman ini sengaja dikosongkan.

Tabel D. 1 Re-Measurement Phase UC Daftar Akun Kontributor

Proses Fungsional	Pergerakan Data	Tipe	CFP
Daftar Akun	1. Aktor memilih tombol “Menu”	E	1
	2. Sistem menampilkan pilihan “ Daftar” dan “Masuk”	X	1
	3. Aktor memilih pilihan “Daftar”	E	1
	4. Sistem menampilkan <i>form</i> “Daftar Akun”	X	1
	5. Aktor mengisi <i>form</i> data profil, yang berisikan “nama akun” , ”kata kunci” , ”konfirmasi kata kunci” dan “peran”	E	1
	6. Aktor memilih peran sebagai kontributor atau <i>reviewer</i>	E	1
	7. Aktor menekan tombol “Lanjut”	E	1
	8. Sistem memvalidasi dan memproses data akun pada database kontributor / <i>reviewer</i>	W	1
	9. Sistem membaca hasil vaildasi data akun pada database kontributor / <i>reviewer</i>	R	1
	10. Sistem menampilkan halaman <i>form</i> “Data Profil”	X	1
	11. Aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap” , “jenis kelamin” , ”tanggal lahir” , ”pendidikan” , ”pekerjaan” dan “alamat email”	E	1
	12. Aktor menekan tombol “Selesai”	E	1
	13. Sistem memvalidasi dan memproses data profil pada database kontributor / <i>reviewer</i>	W	1
	14. Sistem membaca hasil vaildasi data profil pada database kontributor / <i>reviewer</i>	R	1

15. Sistem menampilkan halaman utama kontributor / <i>reviewer</i>	X	1
Alternatif		
Jika memilih sebagai <i>reviewer</i> , maka aktor mengisi semua <i>field</i> pada data profil tentang “nama lengkap”, “jenis kelamin”, ”tanggal lahir”, ”bidang studi”, “pendidikan”, ”pekerjaan”, “alamat email” dan “berkas penunjang”	E	1
Exception		
9.a Sistem membaca hasil vaildasi data akun pada database kontributor / <i>reviewer</i> dan ternyata ada data akun yang tidak valid	R	1
9.b Sistem kembali menampilkan <i>form</i> “Daftar Akun”	X	1
9.c Sistem akan menampilkan peringatan, yang berisi bahwa “Nama Akun Telah Terdaftar” atau “Semua Kolom Harus Diisi”	X	1
14.a Sistem membaca hasil vaildasi data profil pada database kontributor / <i>reviewer</i> dan ternyata ada data profil yang tidak valid	R	1
14.b Sistem kembali menampilkan <i>form</i> “Data Profil”	X	1
14.c Sistem akan menampilkan peringatan, yang berisi bahwa “Semua Kolom Harus Diisi”	X	1

Tabel D. 2 Re-Measurement Phase UC Masuk Akun

	Pergerakan Data	Tipe	CFP
Masuk Akun	Aktor memasukkan “Nama Akun” dan “Kata Kunci” pada form masuk	E	1
	2. Aktor menekan tombol “Masuk” untuk masuk ke dalam sistem	E	1
	3. Sistem mengecek dan memvalidasi data akun dan kata kunci pada database data autentikasi aktor	W	1
	4. Sistem membaca hasil validasi data akun dan kata kunci pada database data autentikasi aktor	R	1
	5. Sistem menampilkan halaman utama aktor	X	1
	Exception		
	4.a Sistem membaca hasil validasi data akun dan kata kunci pada database yang ada pada sistem dan ternyata data akun tersebut tidak valid	R	1
	4.b Sistem kembali ke halaman login	X	1
	4.c Sistem akan menampilkan pesan dibawah kolom nama akun “nama akun tidak sesuai”	X	1

Tabel D. 3 Re-Measurement Phase UC Ubah Detail Akun Pribadi

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Detail Akun	1. Aktor menekan tombol “ubah profil” yang terdapat di bawah data akun lama	E	1
	2. Sistem akan menampilkan halaman	X	1

Pribadi	ubah user dengan kolom yang sudah terisi value data lama		
	3. Aktor mengisi data baru pada kolom yang tersedia	E	1
	4. Aktor menekan tombol “Simpan”	E	1
	5. Sistem memvalidasi dan mengubah data akun pribadi terbaru pada database	W	1
	6. Sistem membaca hasil validasi data akun pribadi terbaru pada database	R	1
	7. Sistem menampilkan” Halaman Akun Pribadi”	X	1
	Exception		
	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem menampilkan “Halaman Akun Pribadi”	X	1
	6.a Sistem membaca hasil validasi data akun pribadi terbaru pada database tersebut dan ternyata data akun pribadi tersebut tidak valid	R	1
	6.b Sistem menampilkan notifikasi bahwa data akun pribadi salah.	X	1

Tabel D. 4 Re-Measurement Phase UC Hapus Akun

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Akun	1. Aktor memilih data akun yang akan dihapus	E	1
	2. Aktor menekan tombol “Hapus” yang terletak disamping nama akun	E	1
	3. Sistem akan menampilkan <i>pop up window</i> berupa pertanyaan “Apakah anda yakin untuk menghapus akun ini?”	X	1
	4. Aktor menekan tombol “Ok”	E	1

	5. Sistem akan menghapus akun dari database	W	1
	6. Sistem akan membaca database akun terbaru	R	1
	7. Sistem kembali ke halaman “daftar akun”	X	1
	8. Sistem menampilkan notifikasi “akun telah berhasil di hapus”	X	1
	Exception		
	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem kembali ke halaman “daftar akun”	X	1

Tabel D. 5 Re-Measurement Phase UC Validasi Kontributor

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi Kontributor	1. Sistem menampilkan pesan ada pengguna baru	X	1
	2. Aktor membuka detail pengguna	E	1
	3. Aktor menekan tombol “Validasi”	E	1
	4. Sistem kembali ke halaman utama admin sistem	X	1

Tabel D. 6 Re-Measurement Phase UC Validasi Reviewer

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi Reviewer	1. Sistem menampilkan pesan ada <i>reviewer</i> baru	X	1
	2. Aktor membuka detail <i>reviewer</i>	E	1
	3. Aktor menekan tombol “Validasi”	E	1

	4. Sistem kembali ke halaman utama admin <i>reviewer</i>	X	1
--	--	---	---

Tabel D. 7 Re-Measurement Phase UC Ubah Kata Kunci

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Kata Kunci	1. Sistem menampilkan halaman dari profil pengguna	X	1
	2. Aktor menekan tombol ubah kata kunci	E	1
	3. Sistem menampilkan halaman untuk ubah kata kunci	X	1
	4. Aktor mengisi field untuk mengubah kata kunci	E	1
	5. Aktor menekan tombol “Simpan”	E	1
	6. Sistem memvalidasi dan mengubah data kata kunci terbaru pada database	W	1
	7. Sistem membaca hasil validasi data kata kunci terbaru pada database	R	1
	8. Sistem kembali menampilkan halaman dari profil pengguna	X	1
	Exception		
	7.a Sistem membaca hasil validasi data kata kunci terbaru pada database tersebut dan ternyata data kata kunci tersebut tidak valid	R	1
	7.b Sistem menampilkan notifikasi bahwa kata kunci salah.	X	1

Tabel D. 8 Re-Measurement Phase UC Tambah Artikel Budaya

Proses Fungsional	Pergerakan Data	Tipe	CFP
Tambah Artikel Budaya	1. Aktor menekan <i>tab menu</i> “Tulis Artikel”	E	1
	2. Sistem akan menampilkan halaman untuk membuat artikel baru	X	1
	3. Aktor menuliskan artikel baru pada <i>field</i> yang tersedia	E	1
	4. Aktor menekan tombol “Simpan”	E	1
	5. Sistem menyimpan artikel baru dari aktor pada database	W	1
	6. Sistem membaca artikel baru dari aktor yang telah disimpan pada database	R	1
	7. Sistem menampilkan halaman “Daftar Master Artikel”	X	1
	Exception		
	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem menampilkan halaman “Daftar Master artikel”	X	1

Tabel D. 9 Re-Measurement Phase UC Ubah Artikel

Proses Fungsional	Pergerakan Data	Tipe	CFP
Ubah Artikel	1. Aktor menekan tombol “Edit Artikel” disamping artikel yang akan diubah	E	1
	2. Sistem akan menampilkan halaman ubah artikel dengan kolom yang sudah terisi <i>value</i> data lama	X	1
	3. Aktor mengisi data baru pada artikel	E	1
	4. Aktor menekan tombol “simpan”	X	1

	5. Sistem akan mengirimkan artikel baru ke database reviewer	W	1
	6. Sistem membaca data artikel yang telah diperbaharui pada database	R	1
	7. Sistem kembali ke halaman “artikel saya”	X	1
	Exception		
	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem kembali ke halaman “artikel saya”Artikel	X	1

Tabel D. 10 Re-Measurement Phase UC Hapus Artikel Budaya

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Artikel Budaya	1. Aktor memilih artikel yang akan dihapus	E	1
	2. Aktor menekan tombol “hapus” yang terletak di samping artikel yang akan dihapus	E	1
	3. Sistem akan menampilkan pop up window berupa pertanyaan “apakah anda yakin untuk menghapus artikel budaya ?”	X	1
	4. Aktor menekan “Ok”	E	1
	5. Sistem akan menghapus artikel dari database	W	1
	6. Sistem membaca database artikel yang telah diperbaharui	R	1
	7. Sistem kembali ke halaman “daftar artikel”	X	1
	8. Sistem menampilkan notifikasi “artikel berhasil dihapus”	X	1
	Exception		

	4.a Jika aktor menekan “Batal”	E	1
	4.b Sistem kembali ke halaman “daftar artikel”	X	1

Tabel D. 11 Re-Measurement Phase UC Cari Artikel Budaya

Proses Fungsional	Pergerakan Data	Tipe	CFP
Cari Artikel Budaya	1. Aktor memasukkan kata / <i>keyword</i> pencarian pada kolom pencarian	E	1
	2. Aktor menekan tombol “Cari Budaya”	E	1
	3. Maka sistem akan menampilkan hasil pencarian	X	1

Tabel D. 12 Re-Measurement Phase UC Unggah File

Proses Fungsional	Pergerakan Data	Tipe	CFP
Unggah File	1. Aktor memilih tombol “ Unggah file budaya”	E	1
	2. Sistem menampilkan halaman “upload”	X	1
	3. Aktor meng- <i>upload file</i> sesuai dengan ketentuan ukuran dan format <i>file</i>	E	1
	4. Aktor menekan “upload”	E	1
	5. Sistem mengunggah file tersebut ke dalam database <i>reviewer</i>	W	1
	6. Sistem membaca file yang telah diunggah pada database <i>reviewer</i>	R	1
	7. Sistem kembali ke halaman “tuliskan artikel”	X	1
	Exception		
	3.a Aktor meng- <i>upload file</i> tidak sesuai dengan ketentuan ukuran dan format <i>file</i>	E	1

	3.b File tidak akan muncul pada file yang di browse Sistem akan menampilkan pesan bahwa ukuran file melebihi batas maksimal	X	1
	5.a Jika aktor menekan “Batal”	E	1
	5.b Sistem kembali ke halaman “tuliskan artikel”	X	1

Tabel D. 13 Re-Measurement Phase UC Validasi Artikel

Proses Fungsional	Pergerakan Data	Tipe	CFP
Validasi Artikel	1. Aktor memilih tombol “ Validasi”	E	1
	2. Sistem menampilkan halaman artikel yang di <i>review</i>	X	1
	3. Aktor menekan tombol “Setujui”	E	1
	4. Sistem akan menambahkan artikel pada database	W	1
	5. Sistem akan menampilkan artikel yang telah divalidasi tersebut	R	1
	6. Sistem menampilkan halaman “list artikel”	X	1
	Exception		
	3.a Jika aktor menekan “Tidak Disetujui”	E	1
	3.b Sistem menampilkan halaman “list artikel”	X	1

Tabel D. 14 Re-Measurement Phase UC Lihat Histori Artikel

Proses Fungsional	Pergerakan Data	Tipe	CFP
Lihat Histori	1. Aktor menekan <i>tab menu</i> “Akitiftas Histori” pada halaman utama kontributor	E	1

Artikel	2. Sistem menampilkan <i>history</i> artikel yang pernah di kirimkan oleh kontributor	X	1
---------	---	---	---

Tabel D. 15 Re-Measurement Phase UC Beri Komentar

Proses Fungsional	Pergerakan Data	Tipe	CFP
Beri Komentar	1. Aktor menekan tab menu “komentar”	E	1
	2. Sistem menampilkan form komentar	X	1
	3. Aktor mengisikan komentar pada kolom “ box komentar”	E	1
	4. Aktor menekan tombol “Kirim”	E	1
	5. Sistem menyimpan komentar dari aktor pada database	W	1
	6. Sistem membaca komentar dari aktor yang telah disimpan pada database	R	1
	7. Sistem menampilkan komentar yang baru dibuat aktor	X	1

Tabel D. 16 Re-Measurement Phase UC Hapus Komentar

Proses Fungsional	Pergerakan Data	Tipe	CFP
Hapus Komentar	1. Aktor memilih menu komentar	E	1
	2. Sistem akan menampilkan komentar-komentar yang terdapat di dalamnya	X	1
	3. Aktor menekan tanda “silang (hapus)”pada komentar yang akan dihapus	E	1
	4. Sistem akan menampilkan halaman pop up yang berisikan “anda yakin akan menghapus komentar ini”	X	1
	5. Aktor menekan “OK”	E	1
	6. Sistem memperbaharui database komentar	W	1

7. Sistem membaca data komentar yang telah diperbaharui pada database	R	1
8. Sistem menampilkan halaman komentar	X	1
Exception		
5.a Jika aktor menekan “Batal”	E	1
5.b Sistem kembali menampilkan halaman komentar	X	1

BIODATA PENULIS



Penulis dilahirkan di kota Surabaya, Jawa Timur. Pada tanggal 7 Mei 1992, merupakan anak kedua dari dua orang saudara. Penulis telah menempuh pendidikan formal hampir diseluruh penjuru Indonesia dan menyelesaikan sekolah dasar hingga sekolah menengah atas di SD GIKI Gubeng Surabaya, SMPN 6 Surabaya, SMAN 2 Surabaya. Tahun 2010 Penulis mendaftar pada Jurusan Sistem Informasi FTIF-ITS dan terdaftar dengan NRP. 5210100049. Di program Studi Sistem Informasi ini Penulis mengambil Bidang Minat Perancangan dan Pengembangan Sistem Informasi (PPSI).

Halaman ini sengaja dikosongkan.